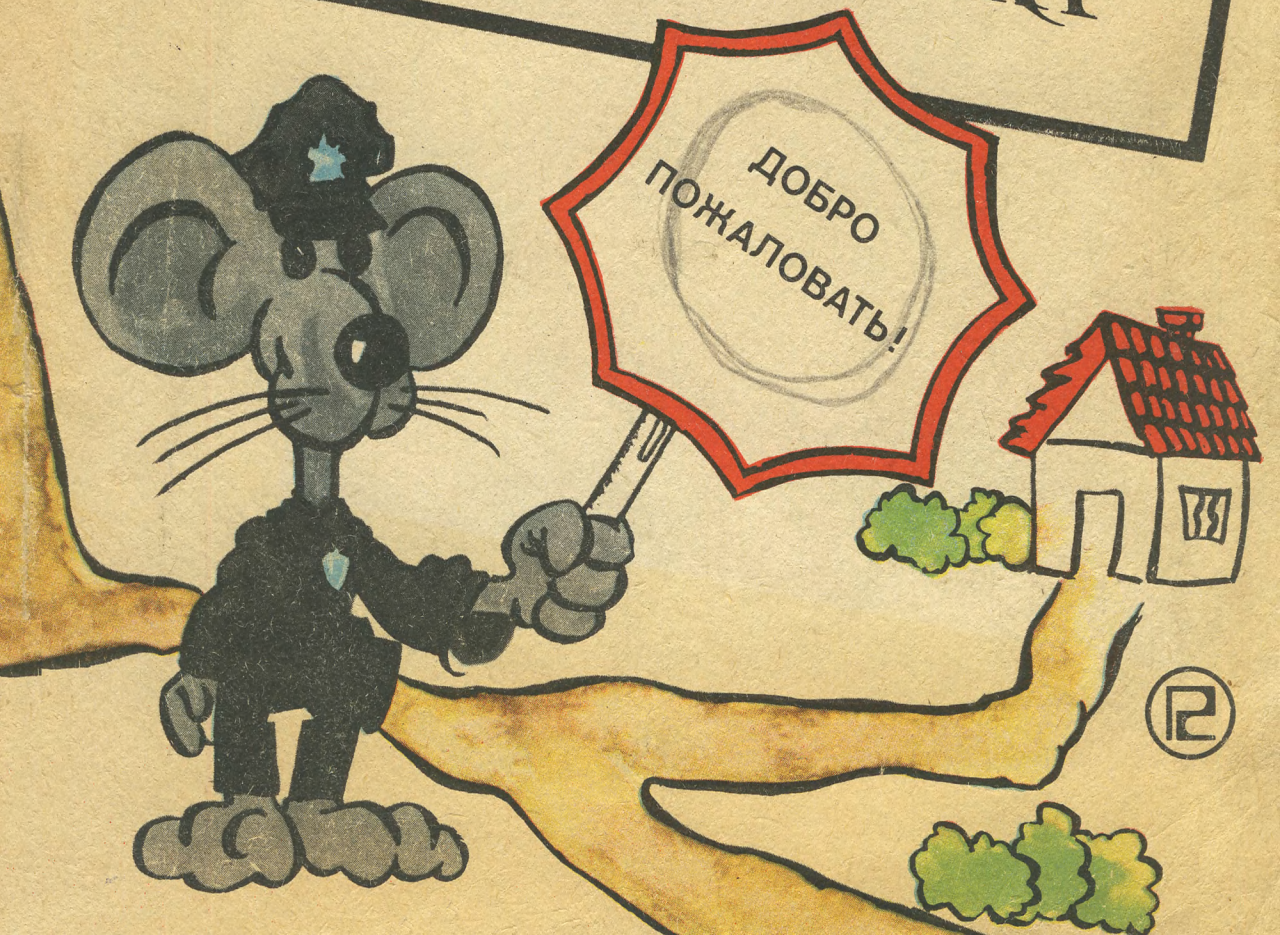
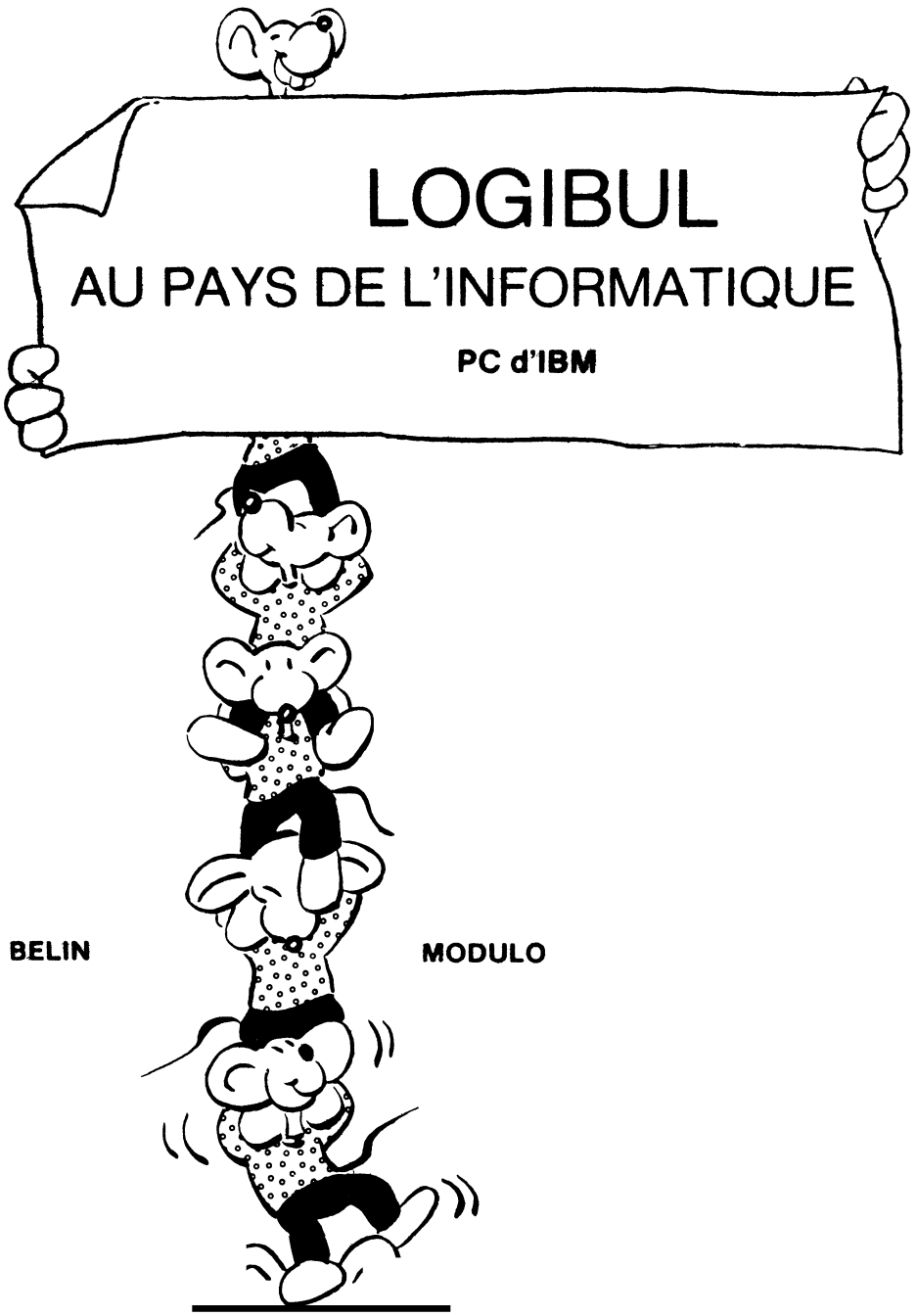
Ш. ДВОРЧИК
Л. ВАСИЛЕНКИ

МЫШКА ПРОГРАММЫШКА В СТРАНЕ ИНФОРМАТИКА



SHEILA DVORCHIK
LESLEY WASYLENKI



LOGIBUL

AU PAYS DE L'INFORMATIQUE

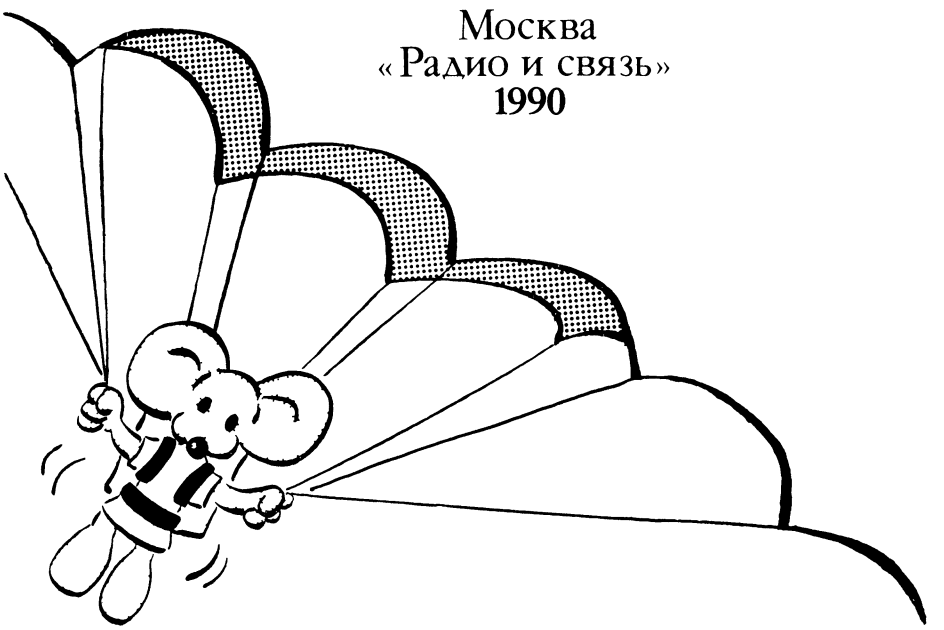
PC d'IBM

BELIN MODULO

Ш. ДВОРЧИК,
Л. ВАСИЛЕНКИ

МЫШКА ПРОГРАММЫШКА В СТРАНЕ ИНФОРМАТИКЕ

Перевод с французского
А. В. Серединского



Москва
«Радио и связь»
1990

ББК 32.973.01
Д24
УДК 519.682.1

Редакция переводной литературы

Дворчик Ш., Василенки Л.

Д24 Мышка Программышка в стране Информатике: Пер. с франц.– М.: Радио и связь, 1990. – 128 с.: ил.

ISBN 5-256-00379-8.

В увлекательной форме книга французского автора знакомит читателей с основными понятиями информатики. Знакомство начинается с элементарных понятий, необходимых для игры и работы с персональным компьютером: клавиатура, курсор, байт, коррекция ошибок и т. д. Затем в игровом представлении описываются основные операторы языка Бейсик, после чего вводятся понятия об алгоритме, символьной информации, случайных числах и их использовании в игровых программах. В конце книги читателю предлагаются разнообразные задания, сложность которых постепенно нарастает.

Для школьников 4–6 классов и преподавателей информатики.

Д $\frac{2404010000\text{-}00}{046(01)\text{-}90}$ 136–90 ББК 32.973.01

Научно-популярное издание

ДВОРЧИК ШЕЙЛА, ВАСИЛЕНКИ ЛЭСЛИ

МЫШКА ПРОГРАММЫШКА В СТРАНЕ ИНФОРМАТИКЕ

Заведующий редакцией **Ю. Г. Ивашов.** Редактор **Л. Ю. Камочкина.** Обложка художника **В. Н. Забайрова.** Художественный редактор **Н. С. Шеин.** Технический редактор **Т. Г. Родина.** Корректор **Л.А. Буданцева**

ИБ № 2021

Сдано в набор 16.08.89. Подписано в печать 18.10.90. Формат 70 × 100 1/16. Бумага тип. № 2. Гарнитура универс. Печать офсет. Усл. печ. л. 10,40. Усл. кр.-отт. 11,38. Уч.-изд. л. 8,77. Тираж 100 000 экз. Изд. № 22839. Зак. №614 Цена 1 р.

Издательство "Радио и связь". 101000 Москва, Почтамт, а/я 693
Набрано в Можайском полиграфкомбинате В/О "Совэкспорткнига" Государственного комитета СССР по печати. г. Можайск, ул. Мира, 93.
Печать и изготовление тиража в Московской типографии №4 Государственного комитета СССР по печати. Москва, 129041, Б. Переяславская, 46

ISBN 5-256-00379-8 (рус.)
ISBN 2-89113-054-5 (франц.)

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ К РУССКОМУ ИЗДАНИЮ

Предлагаемая книга написана в виде пособия по программированию и использованию персонального компьютера. Предполагается, что читатель знакомится с ней, сидя за клавиатурой своей ЭВМ. Читатель, имеющий персональный компьютер типа IBM-PC (отечественные аналоги ЕС-1840, ЕС-1841, «Искра-1030 М»), может использовать все приведенные программы без каких-либо изменений.

Если же читатель имеет дело с компьютером другого типа, то некоторые различия в клавиатуре не должны его смущать: даже фирма IBM выпускает компьютеры с различными вариантами клавиатуры. Авторы книги предусмотрели такой случай: на стр. 6 изображена «пустая» клавиатура, которую читатель может заполнить в соответствии со своим конкретным вариантом.

Если же в распоряжении имеется компьютер другого типа, можно пользоваться книгой как методическим пособием, в котором изложена последовательность овладения навыками программирования и работы за пультом. Занимательные упражнения и примеры могут быть легко воспроизведены на персональном компьютере любого типа.

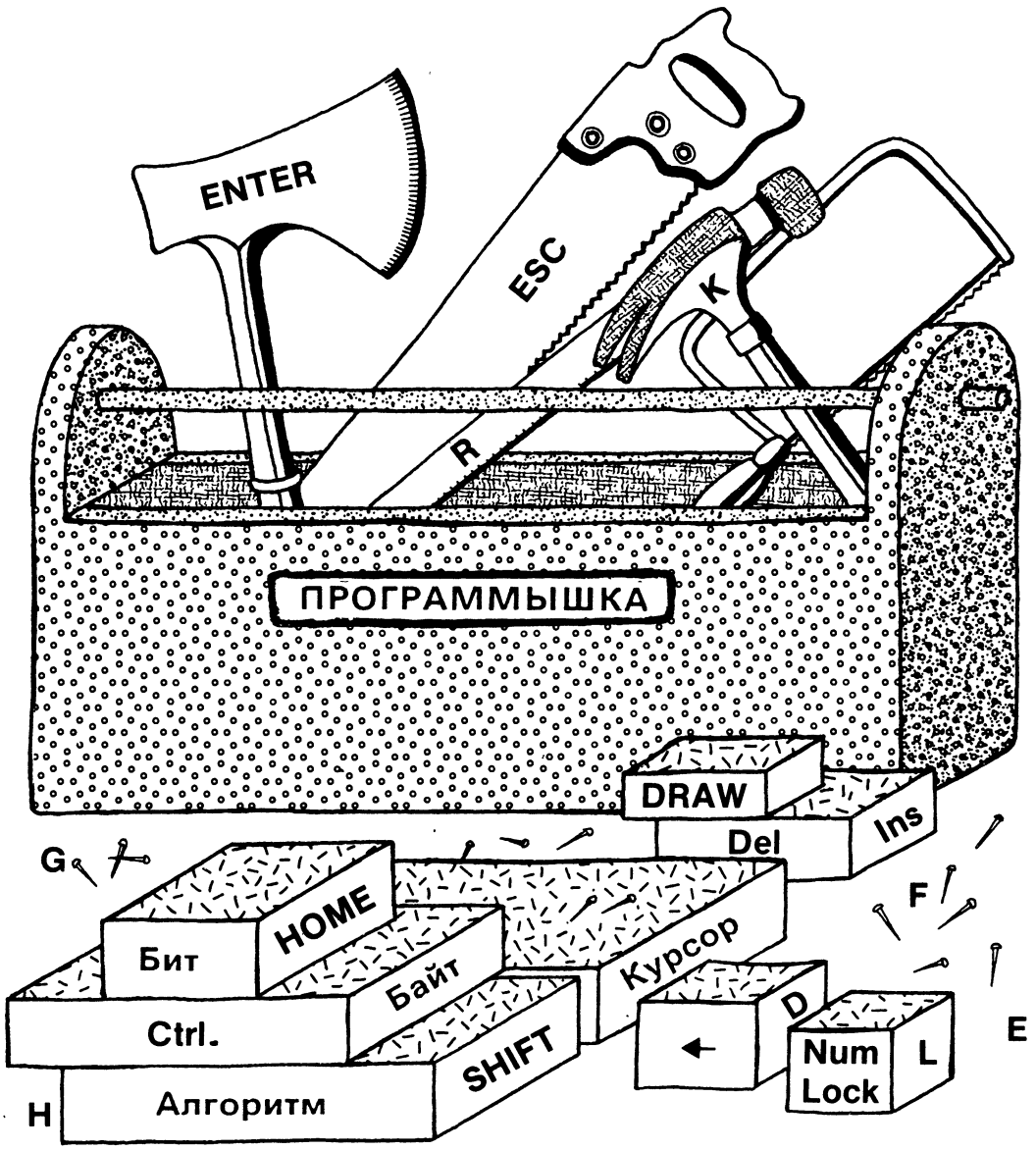
При переводе авторский стиль по возможности сохранен, хотя в отдельных редких случаях пришлось сделать отступления, заменив, например, загадки, скороговорки, имена на более привычные для советского читателя.

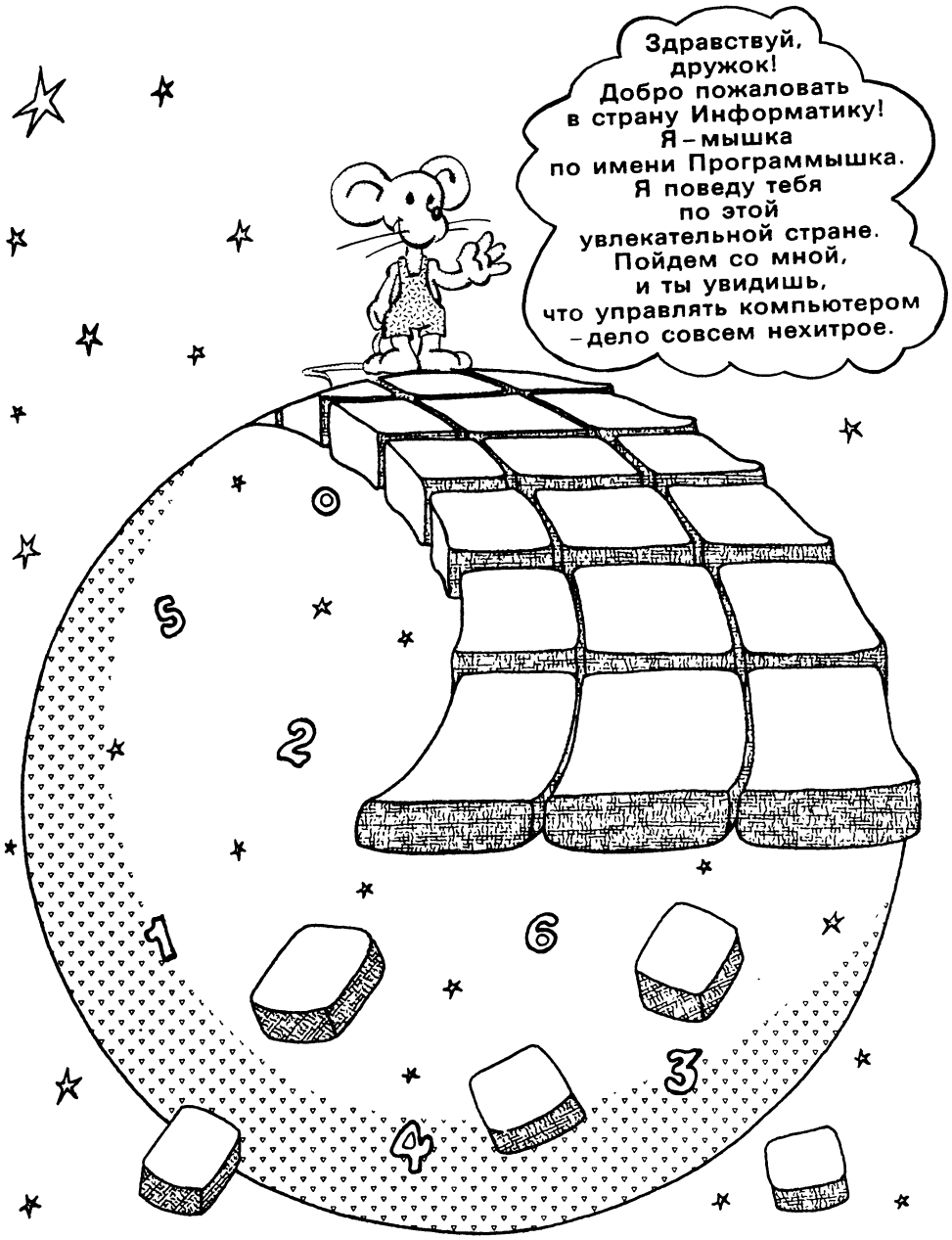
Некоторые из приведенных в оригинале программ проверены нами на персональном компьютере типа „Jamaha".

А. В. Серединский
В. Ф. Жуковский

ПЕРВАЯ ЧАСТЬ

ЯЩИК С ИНСТРУМЕНТАМИ



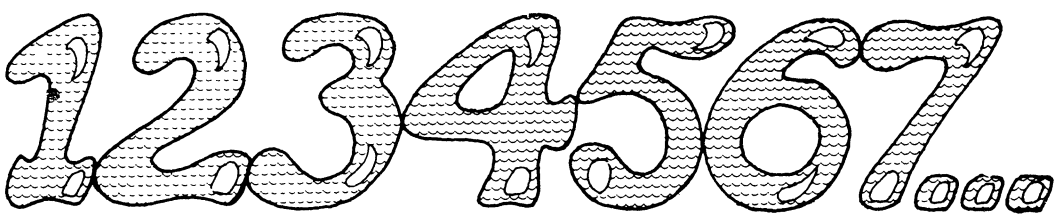

Здравствуй, дружок!
Добро пожаловать в страну Информатику!
Я – мышка по имени Программышка.
Я поведу тебя по этой увлекательной стране.
Пойдем со мной, и ты увидишь, что управлять компьютером – дело совсем нехитрое.
0
5
2
1
6
3
4

Ты ведь, наверное, знаешь, что компьютеру для работы нужна память.

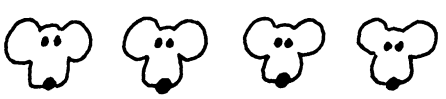
Одна самая маленькая единица памяти называется **БИТ**. Но она так мала, что обычно используют более крупные единицы – **БАЙТЫ** или их половинки – **ПОЛУБАЙТЫ**.

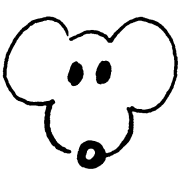
В одном **БАЙТЕ содержится восемь БИТ**, или два **ПОЛУБАЙТА**.

Для записи одной буквы, цифры или символа в компьютере используется один БАЙТ. Поэтому байтов нужно много даже для записи одного слова. Если в твоем компьютере память составляет 64К, то это означает, что в ней содержится около 64 тысяч байт, или 128 тысяч полубайт, или 512 тысяч бит.

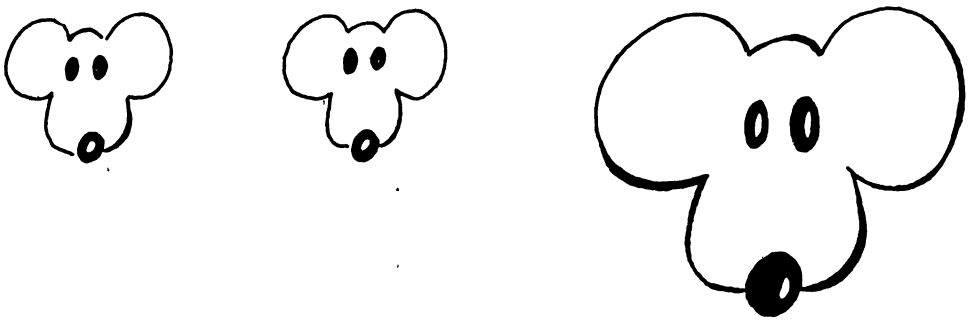


4 бита = **1 полубайт**

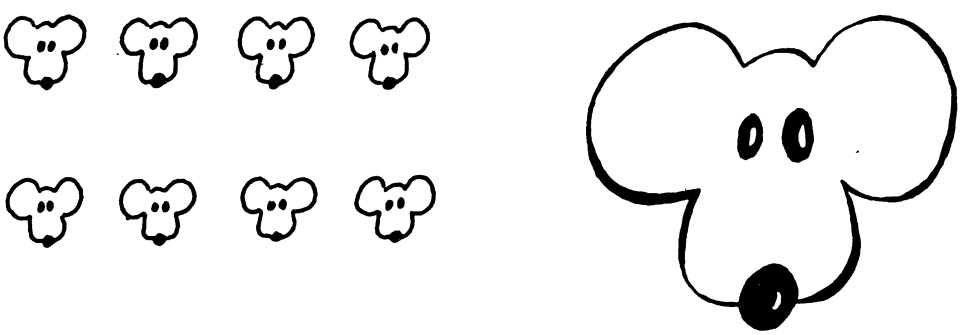




2 полубайта = **1 байт**

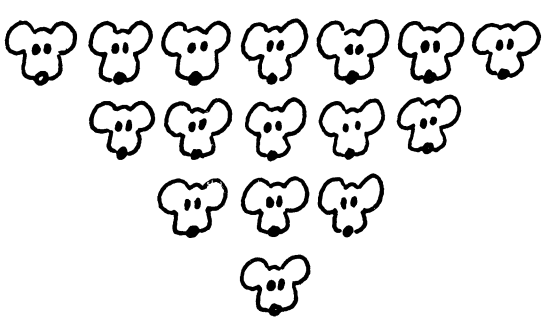


8 бит = **1 байт**

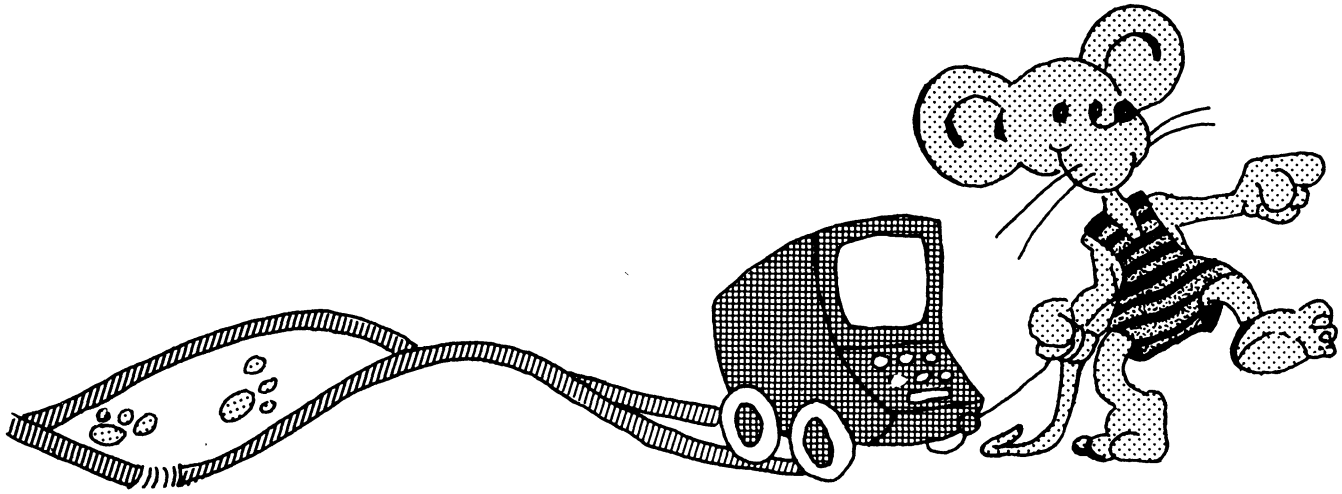


Посмотрим, хорошо ли ты понял мое объяснение.

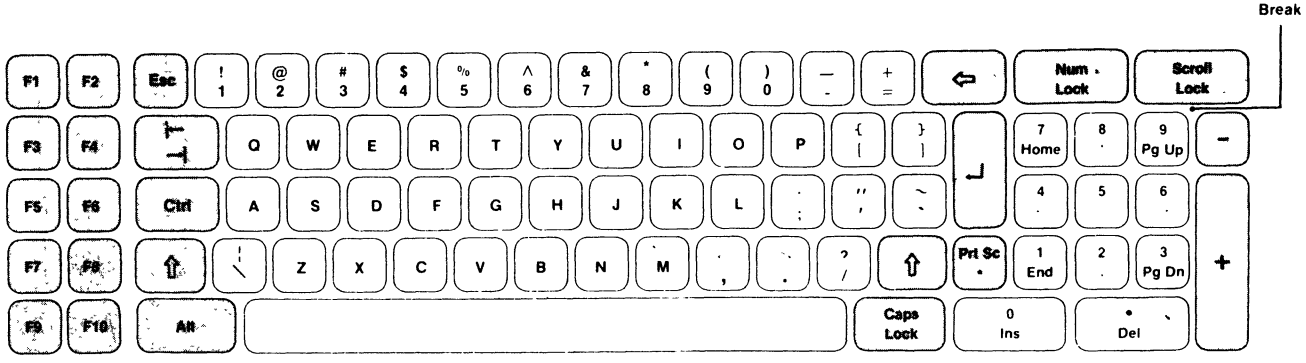
Ну-ка, сосчитай, сколько байт здесь изображено?



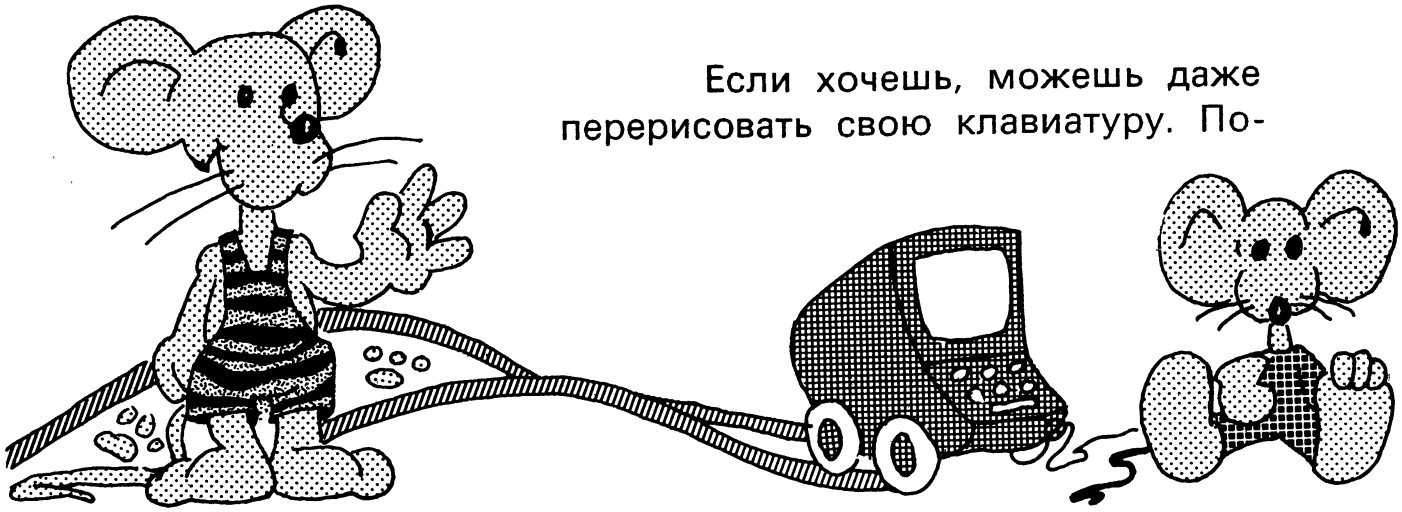
Ответ: 2



Посмотри внимательно на клавиатуру твоего компьютера. Она должна совпадать с одной из двух изображенных здесь клавиатур.

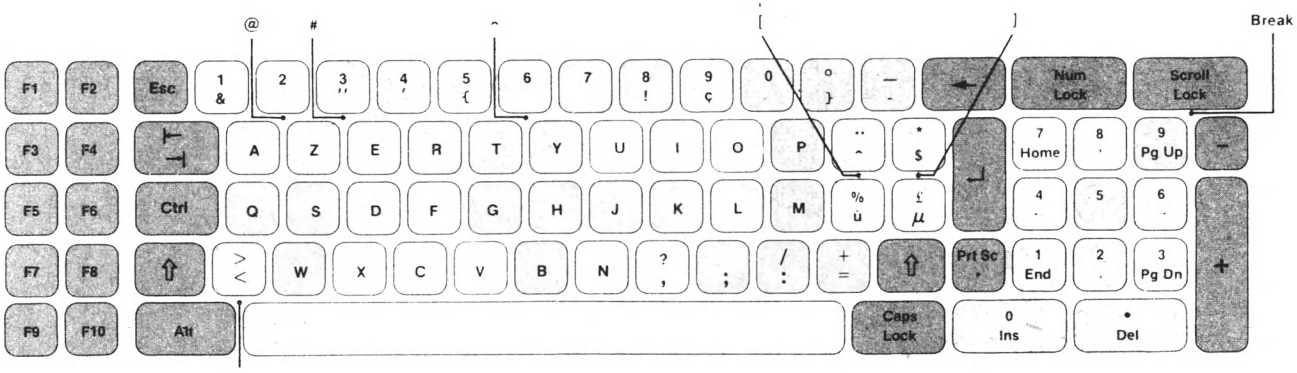


Американская клавиатура

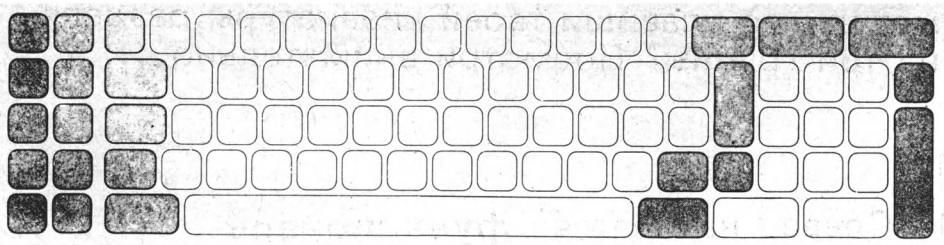


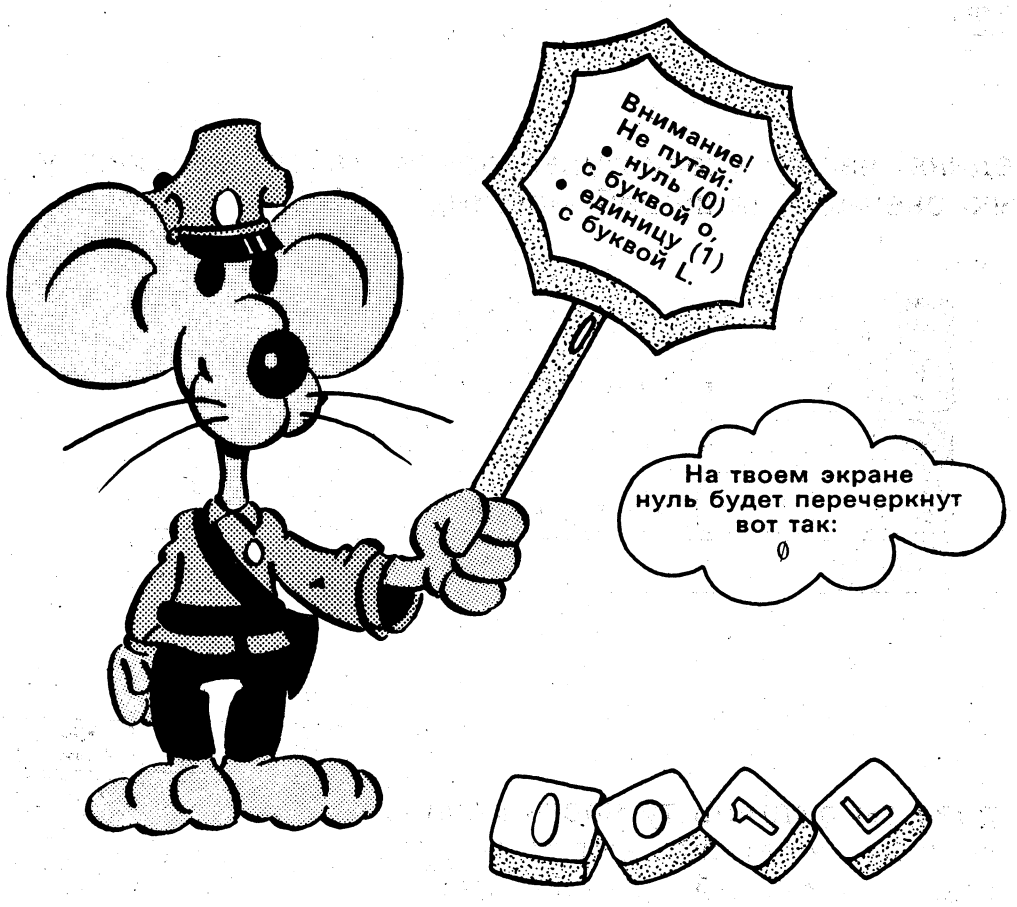
Если хочешь, можешь даже перерисовать свою клавиатуру. По-

смотри на следующую страницу. Там изображена еще одна клавиатура, но она не заполнена. Дорисуй ее так, чтобы она полностью совпадала с твоей. Сначала напиши цифры, потом буквы, затем символы и слова.

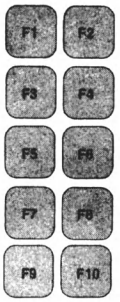


Французская клавиатура



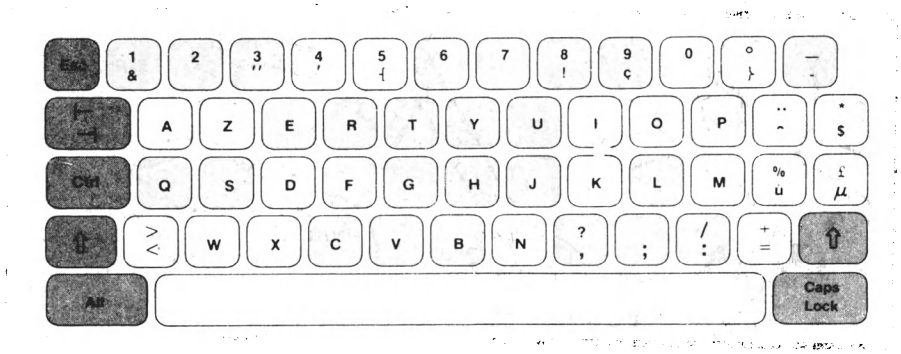

Внимание!
Не путай:
• нуль (0)
с буквой о,
• единицу (1)
с буквой L.
На твоем экране
нуль будет перечеркнут
вот так:
Ø
0
O
1
L

Ты заметил, что клавиши твоей клавиатуры разделяются на три группы? Посмотри внимательно...

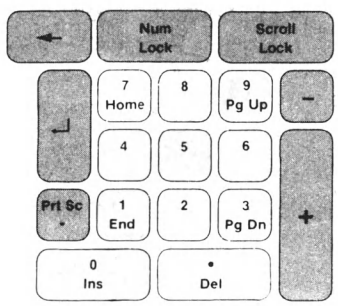


Слева находятся **функциональные клавиши**, представляющие десять *программируемых ключей.* Зачем они нужны – узнаешь немного позже.

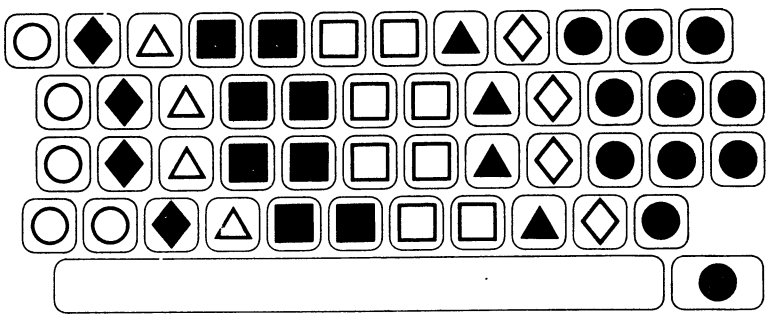
Средняя часть клавиатуры почти такая же, как у обыкновенной пишущей машинки:

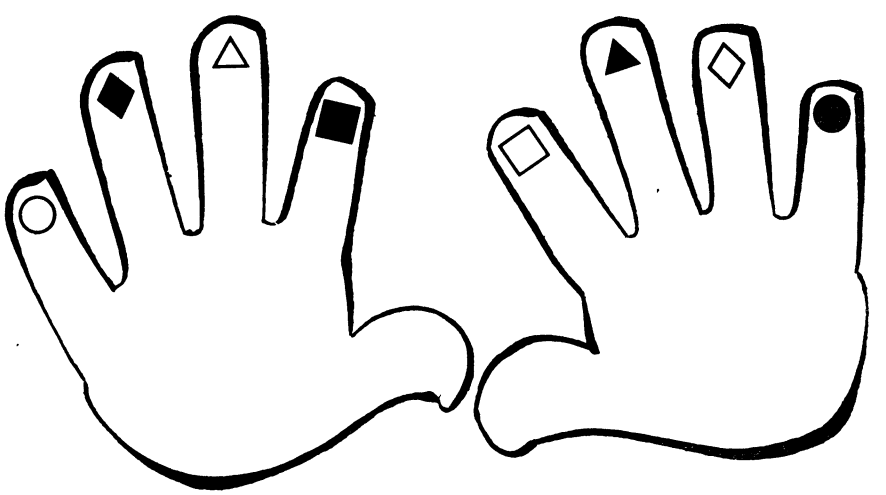


Справа ты видишь ту часть клавиатуры, которая может выполнять несколько функций. Когда нажата клавиша **Num Lock**, белые клавиши используются как **цифровые**.



Чтобы правильно пользоваться центральной частью клавиатуры, ты должен одинаково уверенно действовать всеми пальцами обеих рук. Посмотри на рисунок, и увидишь, какую клавишу каким пальцем следует нажимать.





Для тренировки понажимай клавиши выключенного компьютера. Следи за тем, чтобы каждая клавиша управлялась только нужным пальцем! (Начни с указательного пальца левой руки.) Нажимай только самым кончиком пальца. Вначале тебе будет нелегко, но ты очень быстро освоишься.

Если тебе трудно запомнить, каким пальцем какую клавишу следует нажимать, сделай очень просто: раскрась каждую группу клавиш каким-нибудь цветом, затем приклей к ногтям пальцев кусочки цветной бумаги так, чтобы они совпадали по цвету со своими группами. Немножко потренируйся, а потом выполни следующие упражнения:

1. *Следя за тем, чтобы на каждую клавишу попадал именно тот палец, который требуется,* нажми клавиши в такой последовательности:

A S D F G H J K L : **ПРОБЕЛ**

или, если у тебя французская клавиатура, – в такой последовательности:

Q S D F G H J K L M **ПРОБЕЛ**

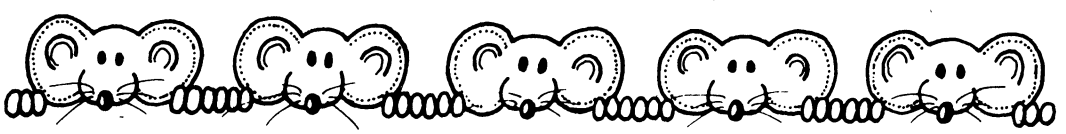
2. Произнеси вслух подряд все буквы алфавита в правильной последовательности, причем в тот момент, когда произносишь название буквы, нажимай на ее клавишу.

3. Нажми подряд клавиши с цифрами от 0 до 9.

4. Напиши: МЫШКА СЪЕЛА ВЕСЬ СЫР.

5. Напиши: УГАДАЙ, ЧТО ЭТО ТАКОЕ: ЗЕЛЕНОЕ, ПОДНИМАЕТСЯ И ОПУСКАЕТСЯ?

6. Напиши: УГАДАЙ, ДЛЯ ЧЕГО ВСЕ МЫШКИ СЕРЫЕ?



Даже если ты не знаешь, для чего предназначены дополнительные клавиши, можешь потренироваться и понажимать их.

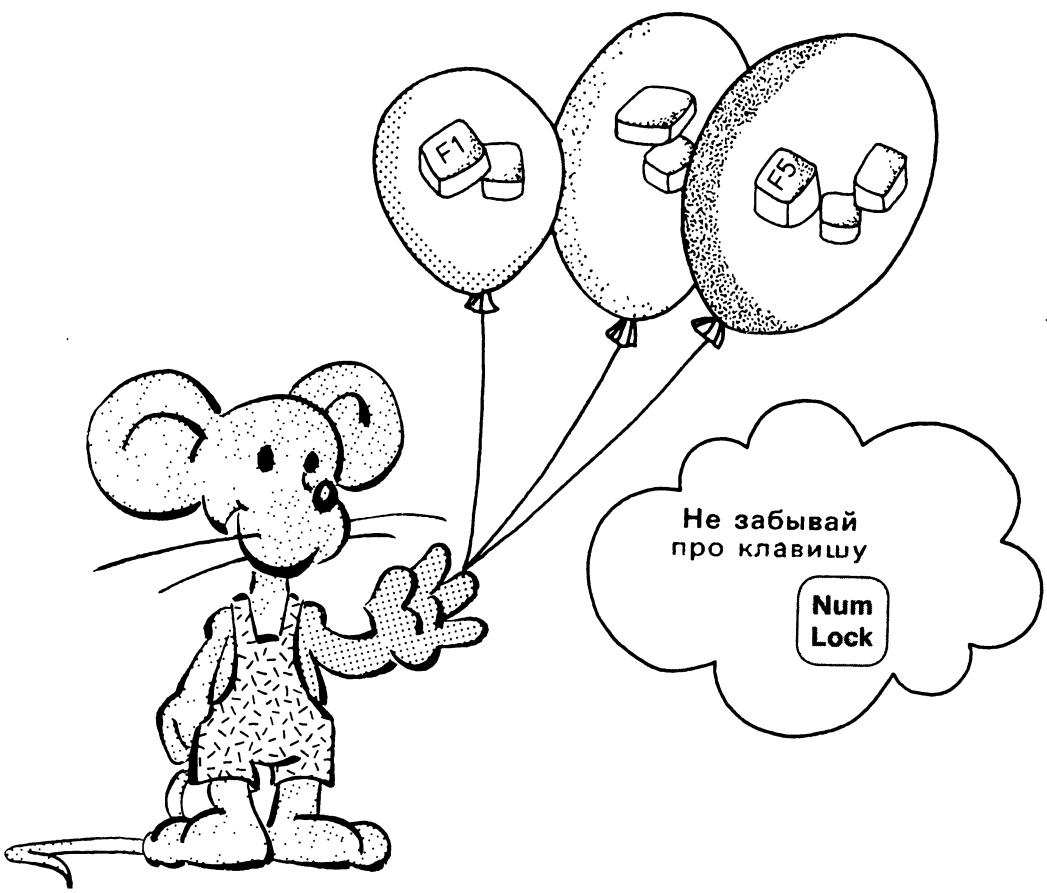
Отгадки. 5. Горошинка в лифте.
6. Чтобы не путать их с розовыми слонами.

Давай договоримся, что цифровые клавиши ты будешь нажимать пальцами только правой руки, а функциональные – только левой. Чтобы тебе было легче, эти группы клавиш специально так и расположены – одна справа, а другая слева от центральной группы клавиш.

Вот еще несколько простых упражнений:

1. Понажимай на клавиши левой группы в такой последовательности: F1, F6, F3, F10, F9, F2, F8, F5, F7, F4.

2. На цифровой клавиатуре набери по порядку все цифры от 0 до 9.

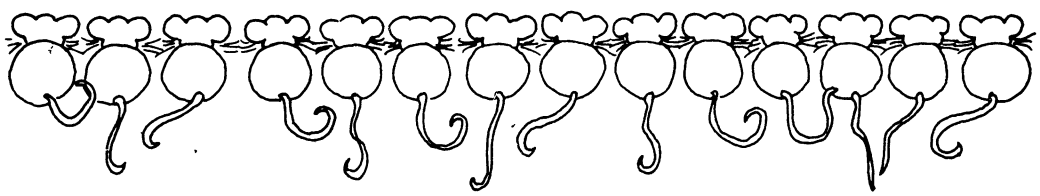


КУРСОР

Теперь, когда ты уже освоился с клавиатурой, включи свой компьютер (выключатель находится сзади справа). Вначале ты услышишь легкое гудение, а потом на экране появится текст, который всегда будет заканчиваться *мерцающей коротенькой черточкой.* Это будет выглядеть примерно так:

```
The IBM Personal Computer Basic
Version C1.10 Copyright IBM Corp
1981
62940 Bytes Free
ok
```

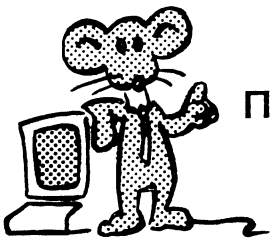
Мерцающая черточка называется **КУРСОР.** Это твой помощник. Ты о нем не забудешь никогда, потому что он будет шаг за шагом неотрывно следовать за тобой.



По мере того как ты будешь писать на экране, верный курсор будет перемещаться и показывать тебе, где появится следующий знак:

```
ПРОГРАМ
```

```
ПРОГРАМ-
МЫШКА
```



Поиграем с клавиатурой

1. Напиши на экране свое имя и нажми на клавишу **ENTER** (↵). Что произойдет? Что делает курсор?

2. Теперь напиши свой адрес. Не забудь шестизначный почтовый индекс! Нажимай на клавишу **ENTER** в конце текста каждой строки.
3. Нажми одновременно на клавишу **SHIFT**(⇧) и на клавишу с наклонной черточкой (/) или, если ты работаешь с французской клавиатурой, – на клавишу с запятой (,). Что произошло? Появился знак вопроса «?»? Ну конечно! Так будет всегда, если ты одновременно с клавишей **SHIFT** будешь нажимать клавишу, на которой изображены два символа; при этом будет появляться тот, который изображен сверху.
4. А ну-ка, попробуй поупражняться с теми знаками, которые сейчас у меня! Если получается, изобрази их на экране. Если же нет, то проделай еще раз внимательно предыдущее упражнение.

СОТРЕМ ВСЕ?

Ты можешь стереть то, что изображено на экране, тремя разными способами. Чтобы освоиться, испробуй их все один за другим. Если на экране ничего нет, то перед каждой проверкой напиши что-нибудь, например свое имя.

☐

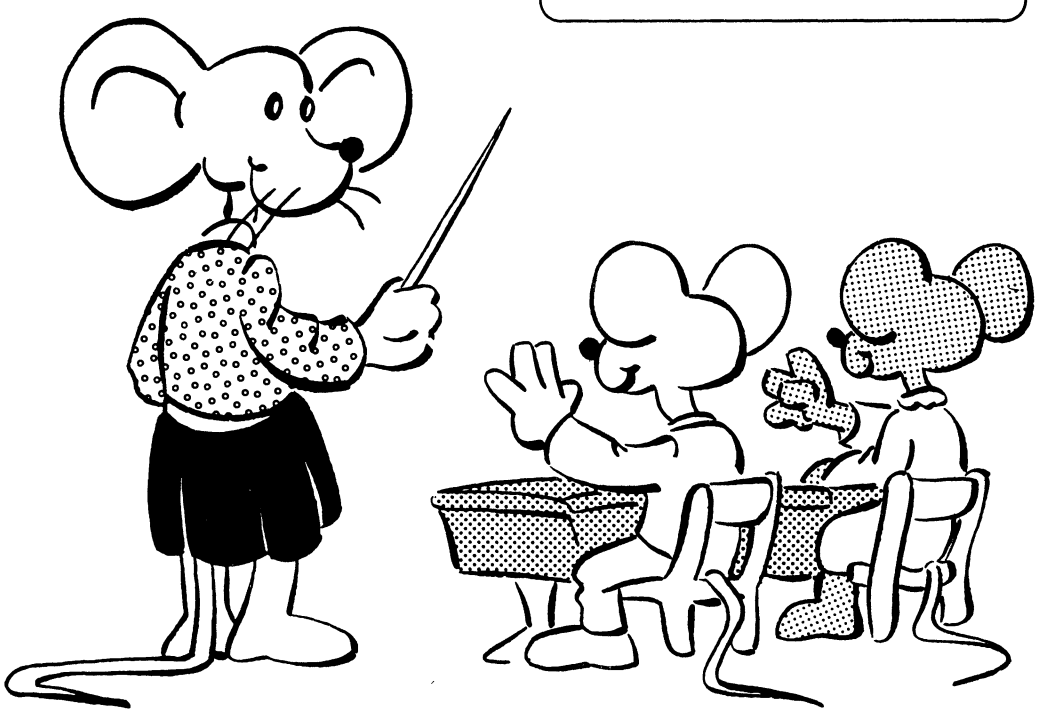
- Нажми одновременно на клавиши CTRL и HOME.

☐☐

- Набери последовательность символов CLS на следующей строке.

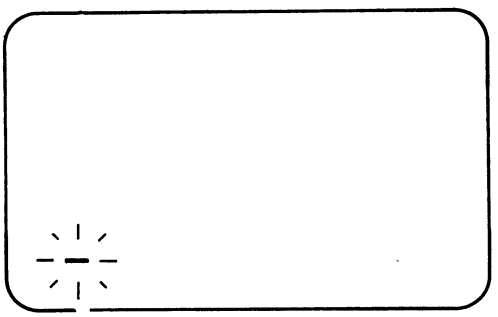
☐☐☐

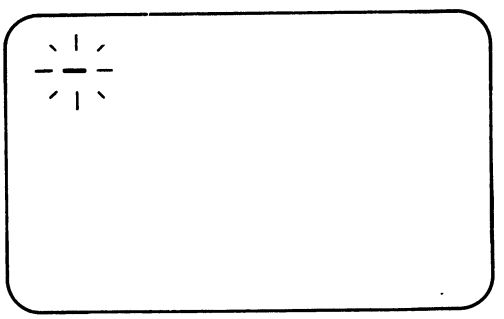
- Нажми на клавишу ENTER (↵)
- Нажми одновременно на клавиши CTRL, ALT и DEL.



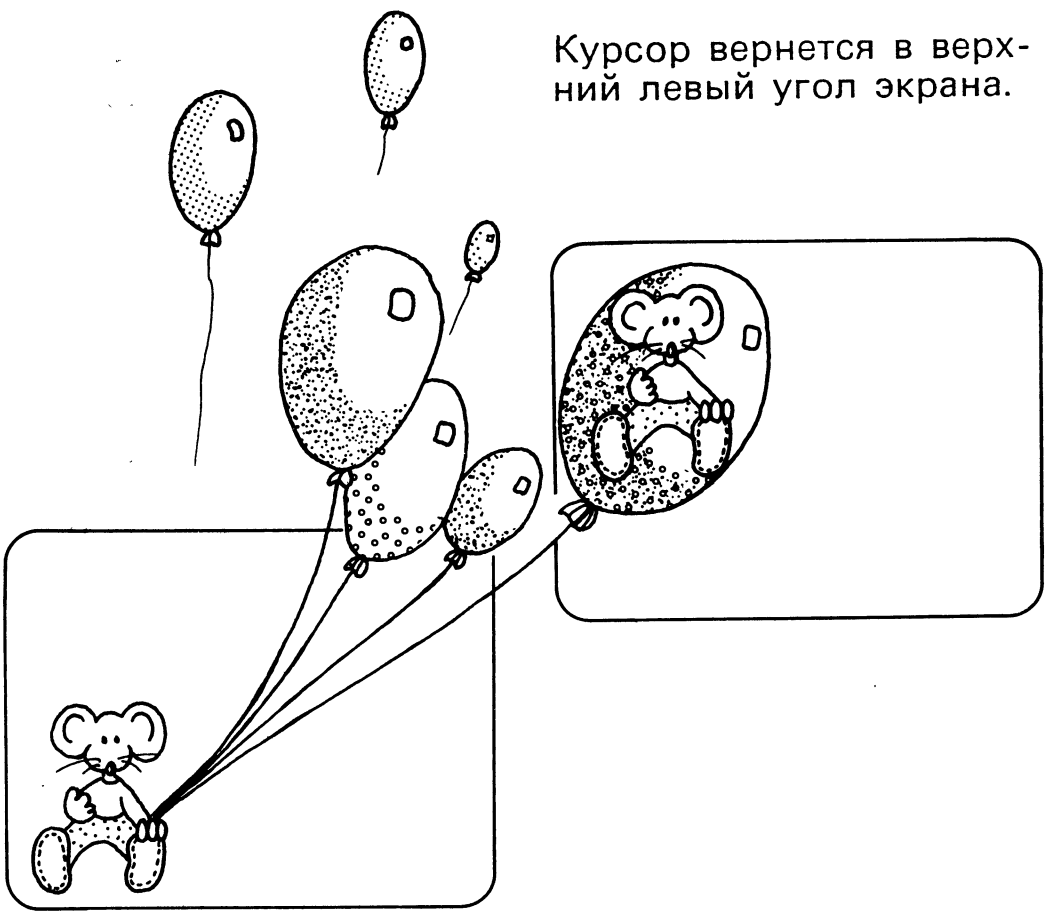
КУРСОР! В УГОЛ!

Нажми на клавишу **HOME** цифровой клавиатуры.



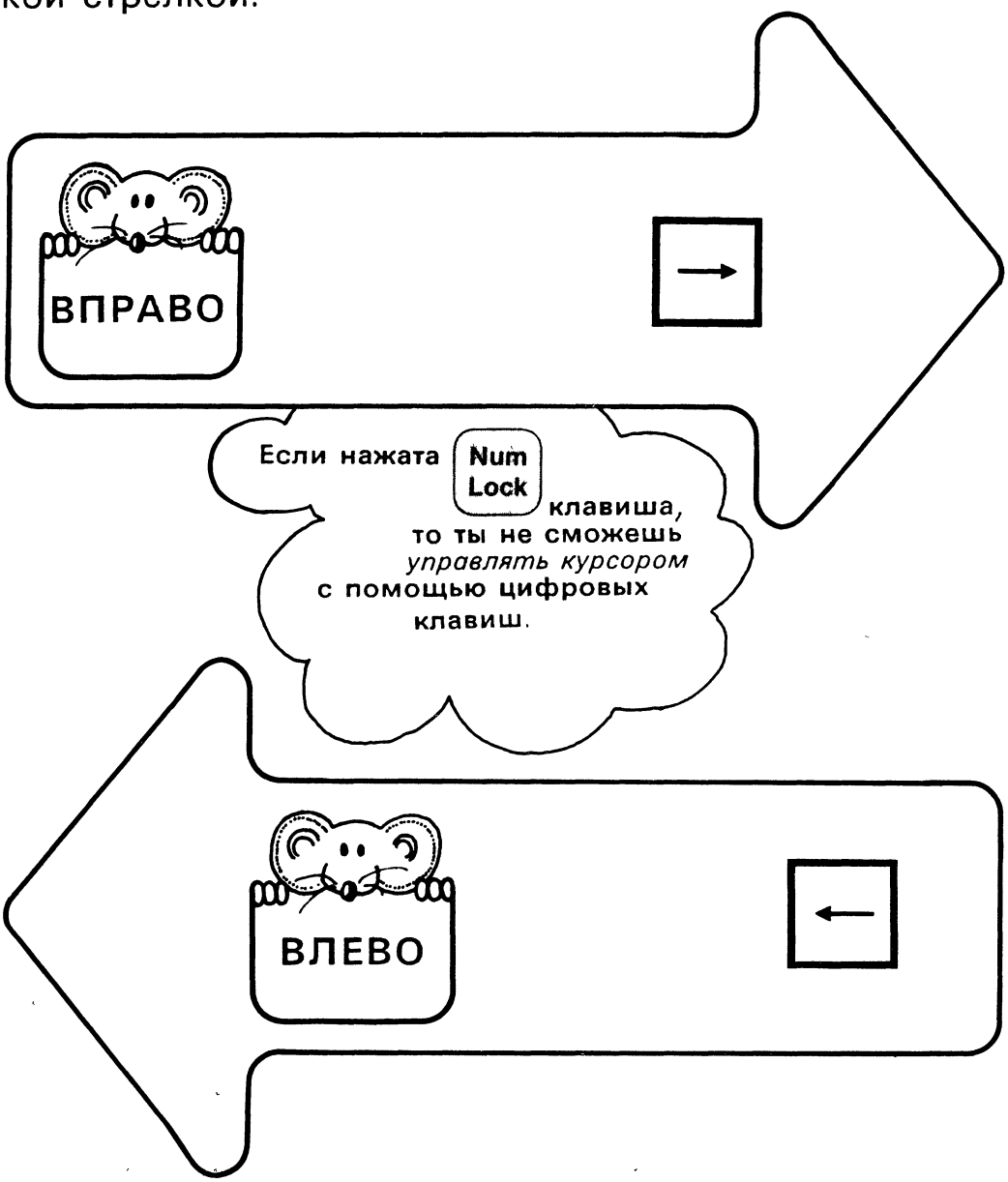


Курсор вернется в верхний левый угол экрана.



КУРСОР ПУТЕШЕСТВУЕТ

Клавиши, которые служат для перемещения курсора, составляют часть цифровой группы клавиатуры твоего компьютера. Каждая клавиша обозначена маленькой стрелкой.



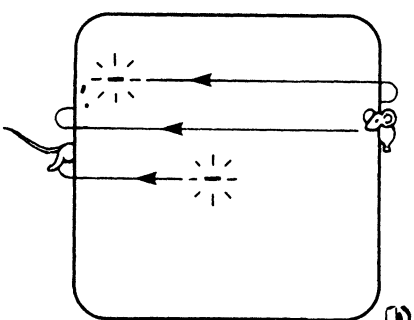
Америка

Передвинь курсор на середину той строки, в которой он сейчас находится.

Нажми на нужную клавишу, чтобы курсор переместился ВЛЕВО.

Продолжай нажимать до тех пор, пока курсор не остановится.
Где он теперь оказался?

Ответ. В углу.

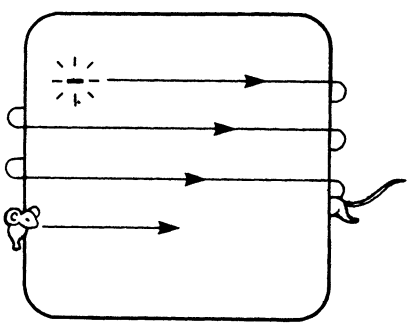


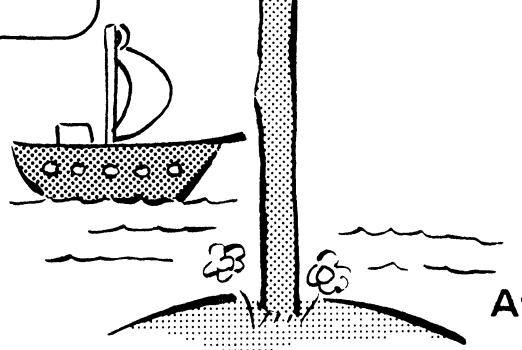
Европа

Нажимай на нужную клавишу, чтобы курсор переместился ВПРАВО.

Что произойдет, когда курсор дойдет до края строки?

Ответ. Перейдет на следующую строку.

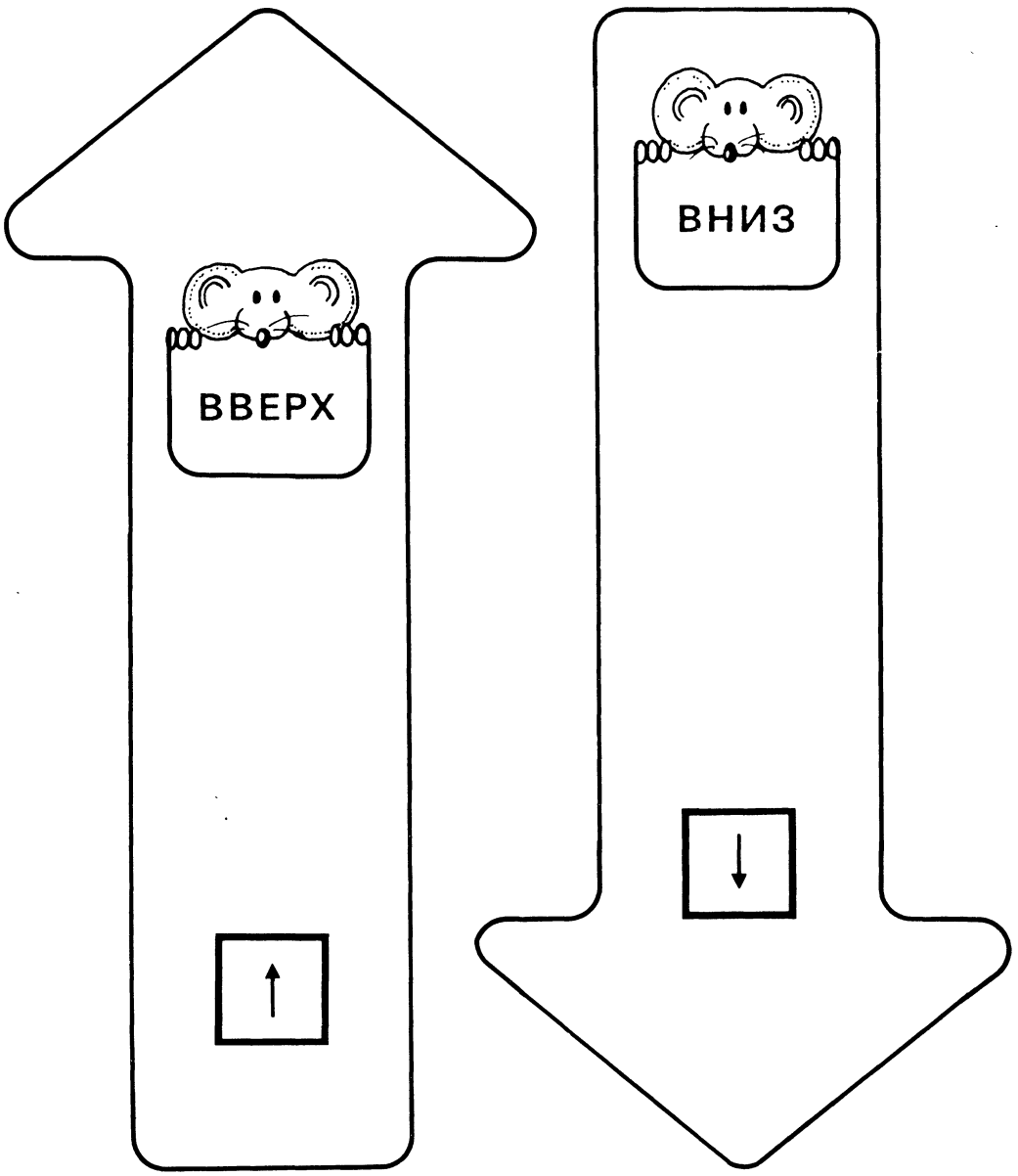


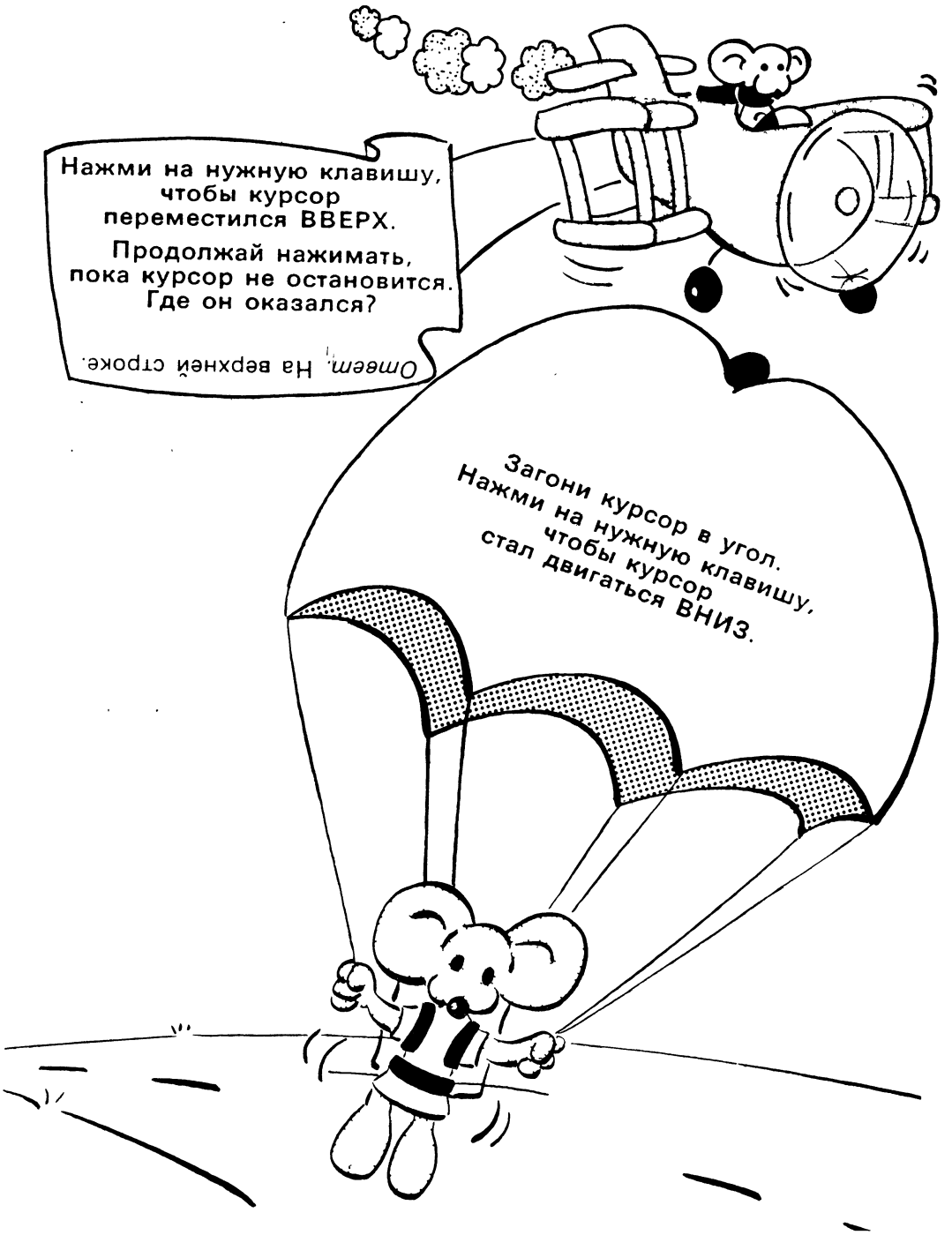


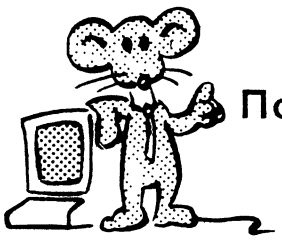
Атлантический океан

1. Что ты делаешь, чтобы передвинуть курсор вправо?
2. Что ты делаешь, чтобы передвинуть курсор влево?

А вот клавиши, которые позволят тебе передвигать курсор **ВВЕРХ** и **ВНИЗ**:




Нажми на нужную клавишу, чтобы курсор переместился ВВЕРХ.
Продолжай нажимать, пока курсор не остановится. Где он оказался?
Ответ: На верхней строке.
Загони курсор в угол. Нажми на нужную клавишу, чтобы курсор стал двигаться ВНИЗ.



Поиграем с курсором

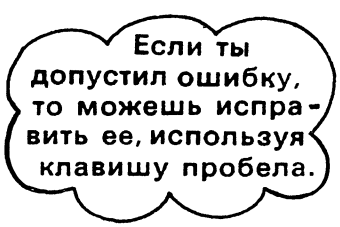
1. Очисти экран.
2. Напиши свое имя в середине экрана.
3. Обведи свое имя рамкой из звездочек (* * * *).
4. Нарисуй вот такое перекрестие:

```
        x
        x
        x
        x
x x x x x x x x x
        x
        x
        x
        x
```

5. А теперь попробуй нарисовать вот такую волну:

```
   ooo         ooo         ooo
 o   o     o   o   o     o   o
o     o   o     o     o   o     o
 o   o     o   o   o     o   o
   ooo         ooo         ooo
```

6. Ну а теперь придумай сам какой-нибудь рисунок и изобрази его. Дай волю своей фантазии!

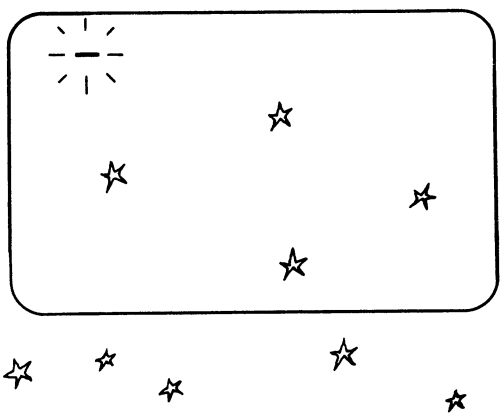


7. Внимательно прочти следующую страницу, а потом поиграй в «Звездную войну».

«ЗВЕЗДНАЯ ВОЙНА»

«Звездная война» – это увлекательная игра, в которую можно играть вдвоем или даже одному. Если ты умеешь перемещать курсор и пользоваться клавишей пробела, то игра не должна вызвать у тебя никаких затруднений.

Цель игры состоит в том, чтобы уничтожить звезды, появившиеся на экране, причем сделать это нужно не позднее чем за 60 секунд. Вот как это делается.



1. Изобрази на экране четыре звезды. Можешь расположить их там, где захочешь, пользуясь клавишами перемещения курсора.

2. Поставь курсор в положение **HOME** (левый верхний угол экрана).

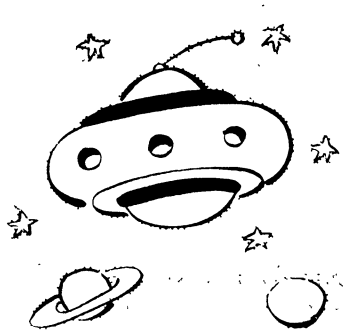
3. Предложи своему сопернику уничтожить расставленные тобой звезды за время, не большее 60 секунд, пользуясь клавишами перемещения курсора и пробела.

4. Заметь время, которое затратил соперник.

5. Попроси соперника расставить четыре звезды, а затем попытайся уничтожить их как можно быстрее.

Выигрывает тот, кто уничтожит чужие звезды быстрее. Чтобы усложнить игру, можно увеличить число звезд или уменьшить время звездного боя.

Если ты играешь один, то поставь себе цель сам и постарайся ее достигнуть.



ИСПРАВЛЕНИЕ ОШИБОК

Ты уже, наверное, успел заметить, что курсор очень полезен. А знаешь ли ты, что он может помочь тебе исправлять ошибки? Посмотри, как это делается. Например, ты мог написать на экране

МОЕГО ДРУГА ЗОВУТ ПРОГРАММИШКА.

С помощью курсора ошибку легко исправить.

1. Поставь курсор под ошибочную букву (**И**):

МОЕГО ДРУГА ЗОВУТ ПРОГРАММИШКА.

2. Нажми на клавишу с нужной буквой (**Ы**):

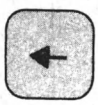
МОЕГО ДРУГА ЗОВУТ ПРОГРАММЫШКА.

3. Отведи курсор за точку, стоящую в конце фразы:

МОЕГО ДРУГА ЗОВУТ ПРОГРАММЫШКА.

А теперь поработай сам. Очисти экран, а затем напиши на нем: ПРОГРАММЫШКА-ХОРОШИЙ МЫШОТОК. Попробуй сам исправить ошибку. Ведь на самом деле я Мышонок, а не Мышоток.
Ответь, что происходит, когда курсор проходит под буквами? Ты прав: НИЧЕГО НЕ ПРОИСХОДИТ.

Существуют и другие клавиши, с помощью которых можно исправлять ошибки. Ты можешь использовать вот эти:


, 
, 
, 
, et 

Клавиши ← и **Del** очень удобны потому, что позволяют убирать лишние буквы одну за другой. У тебя как будто есть два ластика: один правый, другой левый. Вот как надо ими пользоваться:

1. Очисти экран.
2. Напиши **БРРРРРР!**

3. Поставь курсор под последней буквой **Р** и нажми три раза подряд на клавишу ←

БРРРРРР!

БРРРРР!

БРРРР!

БРРР!

3. Поставь курсор под третьей буквой **Р** и нажми три раза подряд на клавишу **Del**.

БРРРРРР!

БРРРРР!

БРРРР!

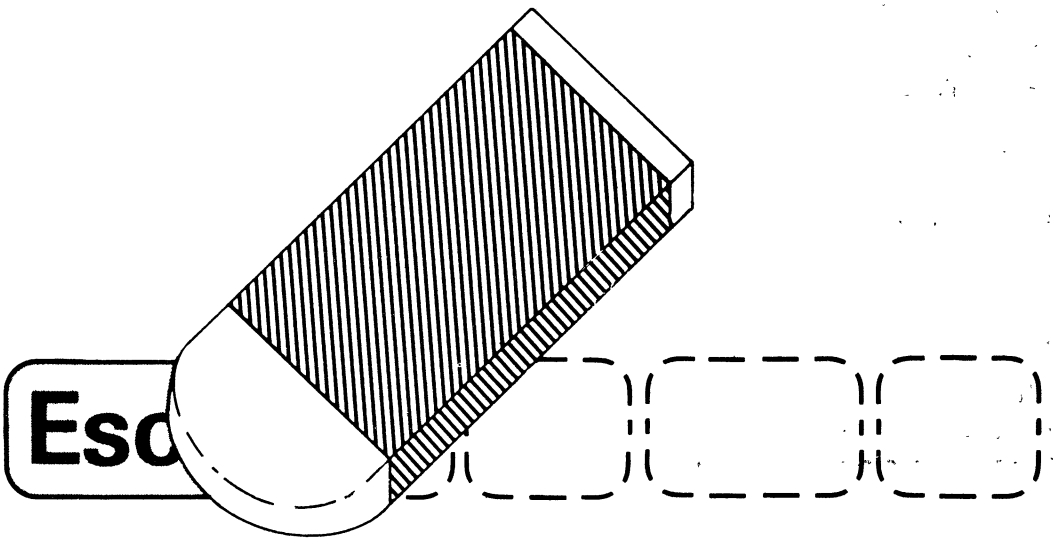
БРРР!

4. Отведи курсор за конец слова.

С помощью клавиш ← и **Del** ты можешь стирать все, что захочешь: одну букву, две буквы, целую строку, параграф и т. д.

Чтобы дело шло побыстрее, не обязательно каждый раз нажимать на клавишу: можешь просто один раз нажать на нее и не отпускать. Только надо быть предельно внимательным, стирание будет происходить очень быстро!

Чтобы хорошенько понять разницу между этими двумя клавишами, поразвлекайся сам, исправляя или изменяя какой-нибудь текст. Например, исправь фразу, которую произносит Мышка Трусишка:



Клавиша **Esc** тоже служит для стирания текста, но она гораздо «прожорливее», чем другие. Давай проверим.

1. Очисти экран.
2. Напиши: **Esc** – БОЛЬШАЯ ОБЖОРА.
3. Нажми на клавишу **Esc**.

Что произошло? Вся строка исчезла? Ну конечно! Так будет каждый раз, как только ты нажмешь на клавишу **Esc**. Когда понадобится, ты можешь этим воспользоваться.

Когда ты пишешь на бумаге, ошибку всегда можно исправить с помощью ластика. Если ты что-то пропустил, сотрешь то, что уже написано после ошибки, исправишь и напишешь заново.
Компьютер дает тебе возможность исправлять любые ошибки, вставлять буквы, слова и даже целые фразы, ничего не стирая. Смотри внимательно:

1. Напиши фразу.

МНЕ НРАВИТСЯ ИНФОРМАТИКА.

2. Подведи курсор под начальную букву последнего слова.

МНЕ НРАВИТСЯ ИНФОРМАТИКА.

3. Нажми клавишу **Ins**. Курсор при этом исчезнет, а буква, под которой он находился, начнет мерцать.

МНЕ НРАВИТСЯ ИНФОРМАТИКА.

4. Напиши слово **СТРАНА** и после него поставь пробел.

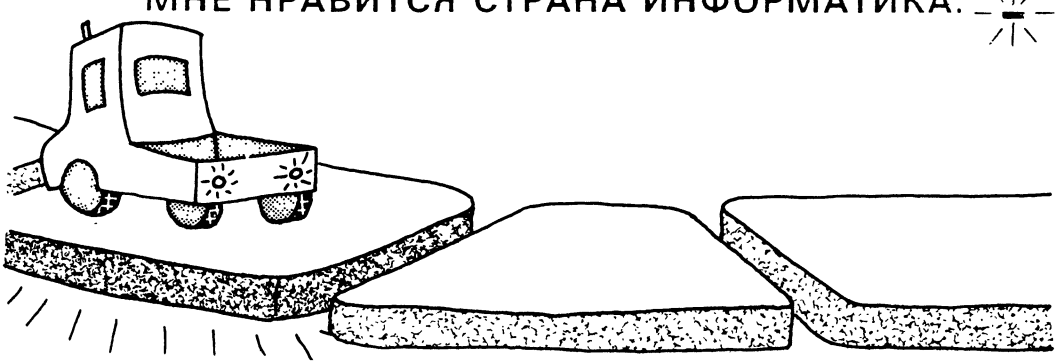
МНЕ НРАВИТСЯ СТРАНА ИНФОРМАТИКА.

5. Снова нажми на клавишу **Ins**, и мерцание прекратится, а курсор опять появится.

МНЕ НРАВИТСЯ СТРАНА ИНФОРМАТИКА.

6. Помести курсор после точки в конце фразы.

МНЕ НРАВИТСЯ СТРАНА ИНФОРМАТИКА.



END

Чтобы поместить курсор в конец строки, можешь использовать клавишу **End**.

1. Снова набери фразу из предыдущего упражнения.

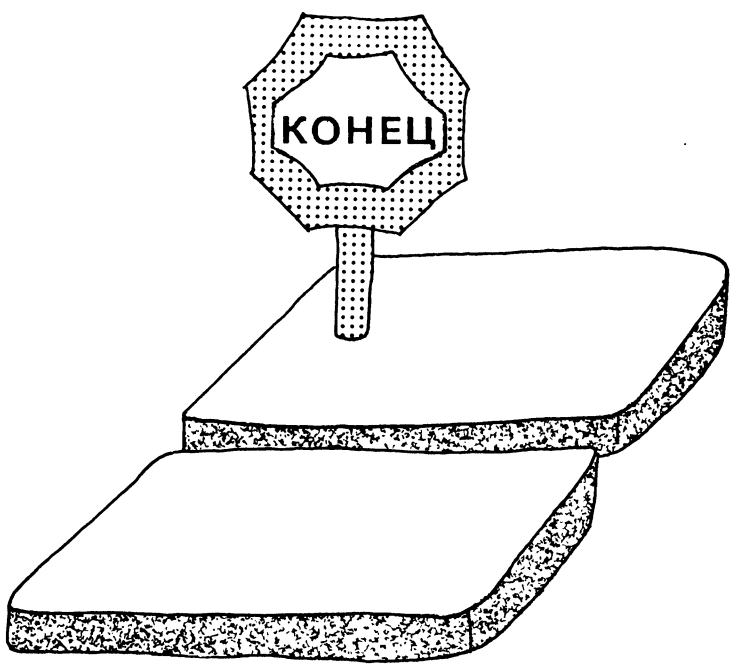
МНЕ НРАВИТСЯ СТРАНА ИНФОРМАТИКА.

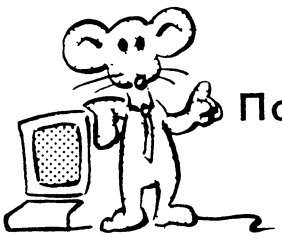
2. Сотри слово **СТРАНА** с помощью клавиш ← или **Del**, а потом установи курсор в конец строки, используя клавишу **End**.





Теперь у тебя есть все инструменты, необходимые для исправления текста на экране. Действуй!





Поиграем с клавишами

1. Очисти экран.

2. Напиши: **ИНТЕРЕСНО ИГРАТЬ С КОМПЬЮТЕРОМ**. Потом добавь слово **ЭТИМ** после предлога **С**.

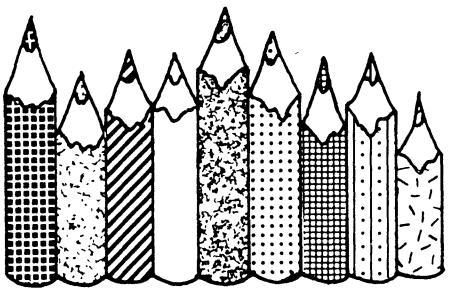
3. Нарисуй такую картинку:

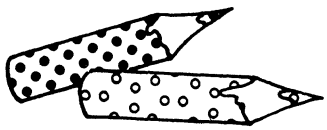
```
x x x x
x     x
x     x
x x x x
```

4. Теперь нарисуй вот такую картинку, а потом преврати прямоугольник в квадрат с помощью клавиши **Ins**:

```
    x           x
x x x x x x x x x
    x           x
    x           x
x x x x x x x x x
    x           x
```

ПОРИСУЕМ





После серьезной работы, которую мы с тобой проделали, небольшая разрядка будет вполне заслуженной, не правда ли? Пойдем со мной. У меня есть друзья, с которыми я хочу тебя познакомить.

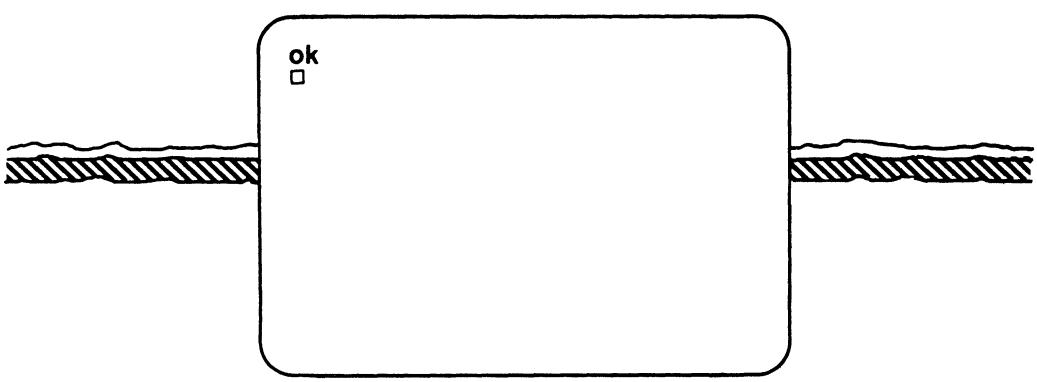
Впрочем, надо подготовить к их приходу экран:

1. Напиши: **SCREEN 1** и нажми на клавишу **ENTER** (↵).

2. Напиши: **KEY OFF** и нажми на клавишу **ENTER** (↵).

3. Напиши: **CLS** и нажми на клавишу **ENTER** (↵).

Если ты точно выполнил все мои советы, то на экране должно появиться вот что:



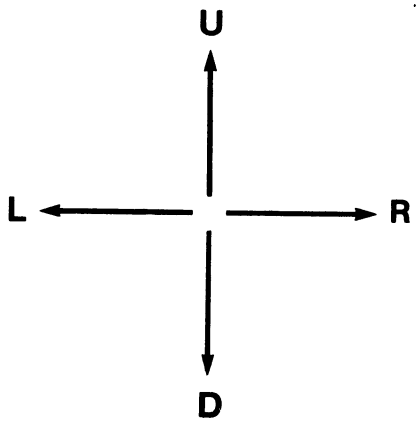
Друзья, о которых я тебе говорю, – это группа художников, которых никто никогда не может увидеть. Однако они всегда на месте. У них не совсем обычные имена, но ты к ним быстро привыкнешь. У каждого художника своя специальность:

U рисует вверх (*up*),

D рисует вниз (*down*)

L рисует влево (*left*),

R рисует вправо (right).



Чтобы вызвать каждого художника, достаточно написать **DRAW** и указать, сколько шагов ему следует пройти. Перепиши без ошибок эти данные:

```
DRAW "U25
DRAW "R100
DRAW "D75
DRAW "L75
DRAW "U50
DRAW "R50
DRAW "D25
DRAW "L25
```

Тогда на экране получишь

```
ok
draw "u25
ok
draw "r100
ok
draw "d75
ok
draw "l75
ok
draw "u50
ok
draw "r50
ok
draw "d25
ok
draw "l25
ok
□
```

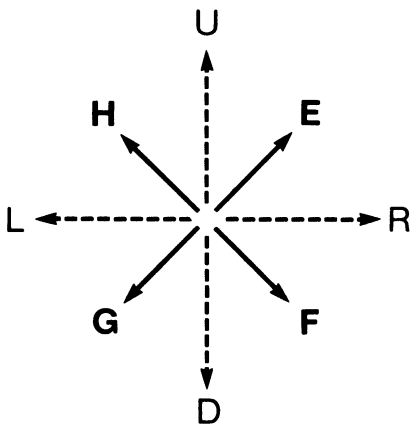
Чтобы получить наклонные линии, нужно вызвать художников **E, F, G** и **H**:

E рисует вверх вправо,

F рисует вниз вправо,

G рисует вниз влево,

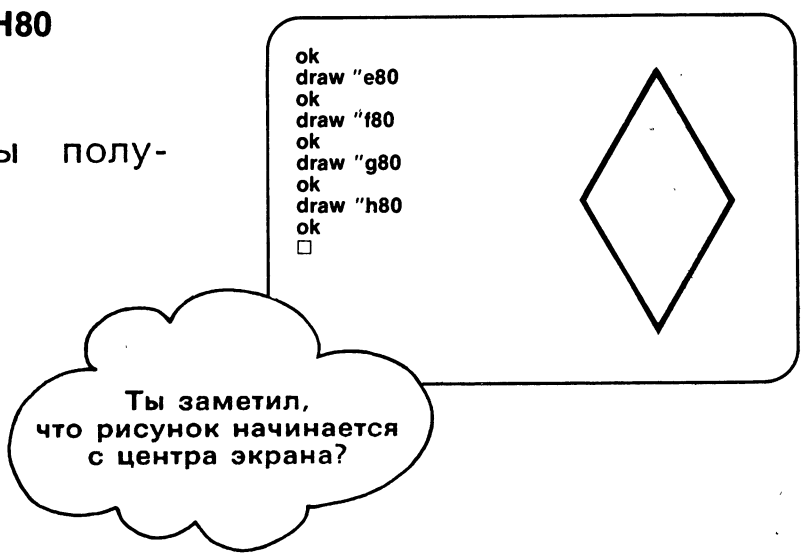
H рисует вверх влево.



Очисти экран (**CLS**), перепиши эти данные, а затем напиши команду **RUN**:

```
DRAW "E80
DRAW "F80
DRAW "G80
DRAW "H80
```

Тогда ты получишь

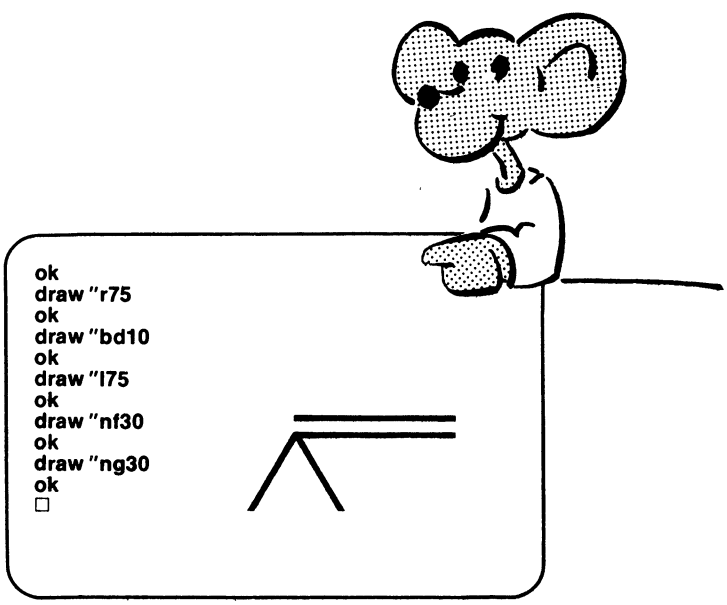


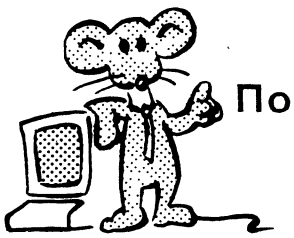
Ну вот, наконец, два последних члена команды **B** и **N**. В отличие от других они не могут работать поодиночке, а только в паре с кем-нибудь из художников. Ты обязательно должен давать каждому из них по компаньону.

B делает рисунок другого художника невидимым.
N заставляет художника вернуться в начальную точку после того, как он нарисует свой отрезок.

Попробуй, например, выполнить вот эти инструкции:

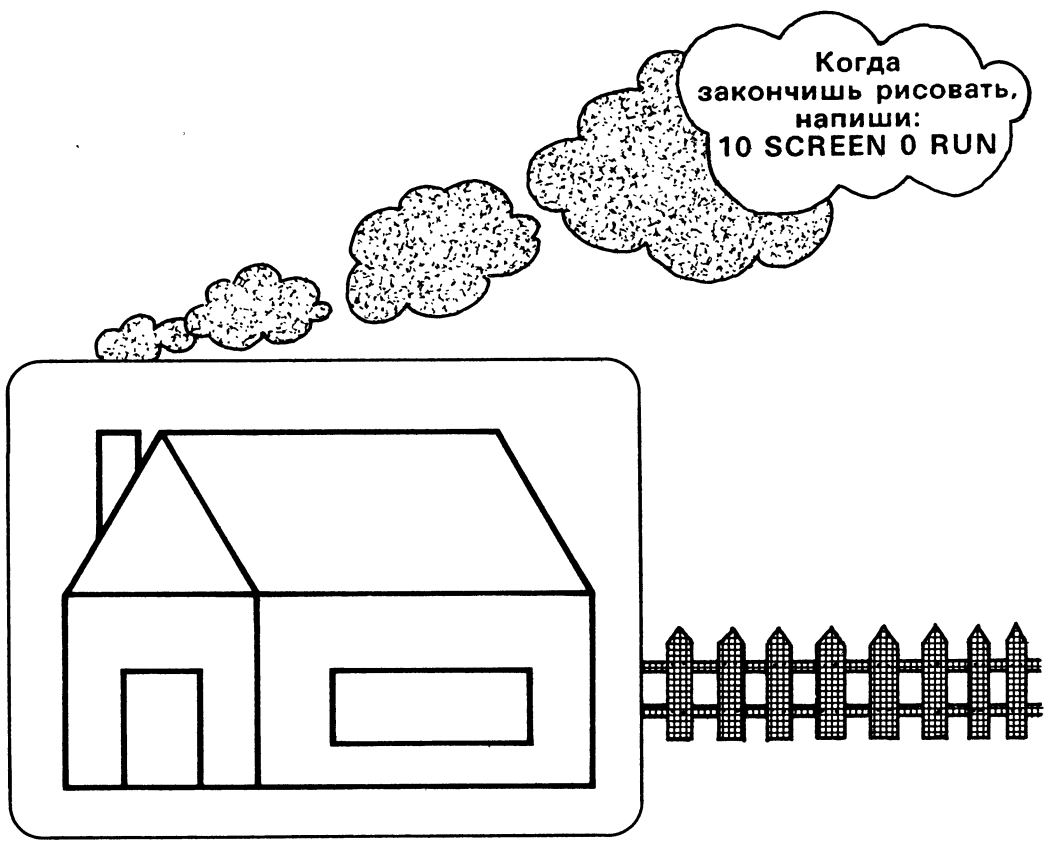
DRAW "R75 — отрезок длиной 75 единиц вправо.
DRAW "BD10 — невидимый отрезок длиной 10 единиц вниз.
DRAW "L75 — отрезок длиной 75 единиц влево.
DRAW "NF30 — наклонный отрезок вниз вправо длиной 30 единиц и возвращение в исходную точку.
DRAW "NG30 — наклонный отрезок вниз влево длиной 30 единиц и возвращение в исходную точку.





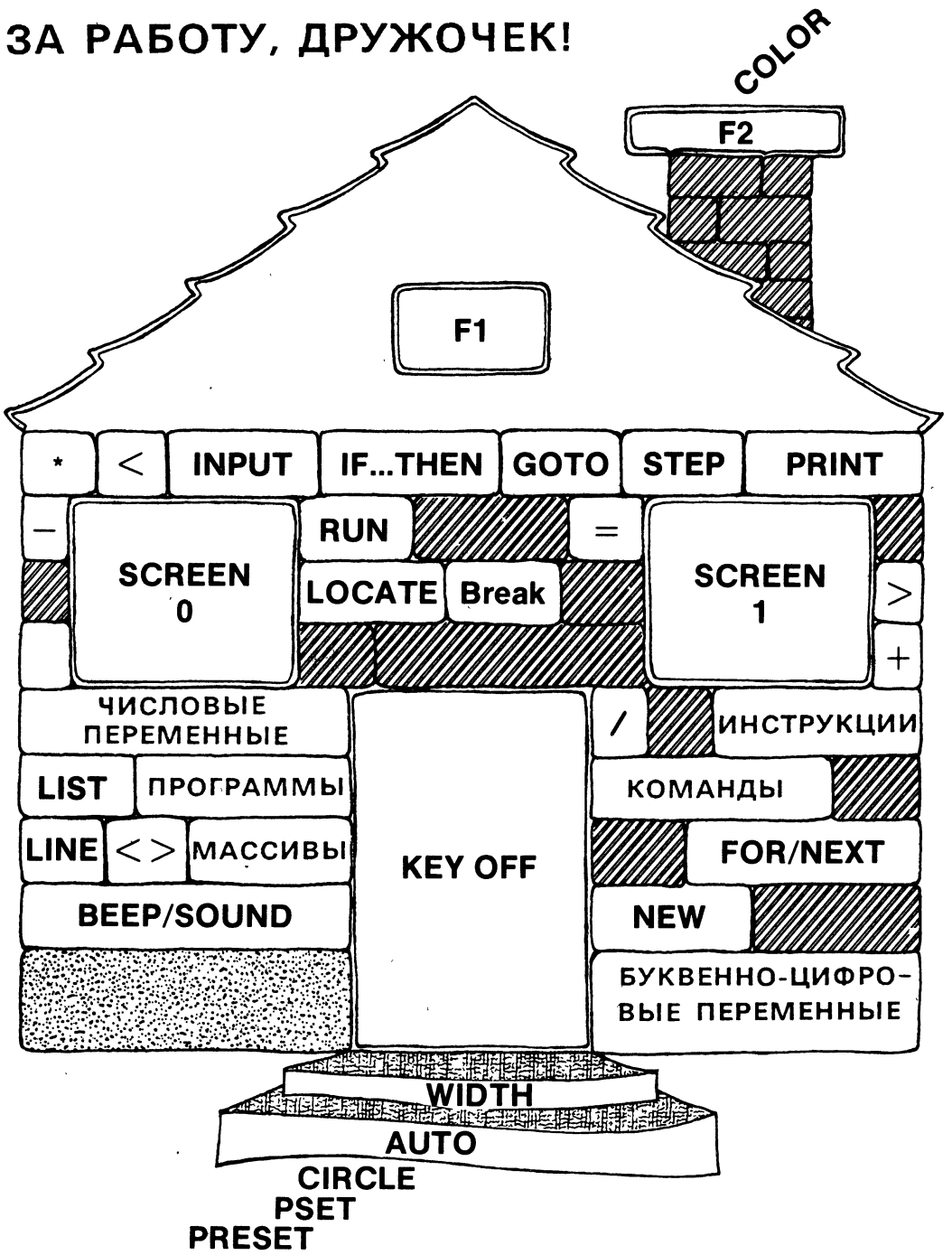
Поиграем с отрезками

1. Очисти экран (**CLS**).
2. Нарисуй квадрат со стороной 20 единиц.
3. Нарисуй восьмиконечную звезду с лучами длиной по 10 единиц каждый.
4. Попробуй нарисовать вот такой домик:



ВТОРАЯ ЧАСТЬ

ЗА РАБОТУ, ДРУЖОЧЕК!



До сих пор мы с тобой только развлекались, а теперь пора приступать к серьезным делам. Чтобы по-настоящему проникнуть в страну Информатику, надо как следует поработать. Самое главное – хорошо знать ее язык и соблюдать все ее законы.

Жители города, в котором мы с тобой сейчас находимся, говорят на языке БЕЙСИК. Этот язык тебе предстоит выучить. Не волнуйся, это совсем не так сложно, ведь речь идет о простом языке программирования, в котором очень мало слов.

PRINT

PRINT – это одно из самых главных слов в языке БЕЙСИК. Если ты хочешь, чтобы компьютер изобразил что-нибудь на экране, то обязательно должен воспользоваться именно этим словом. Но нельзя забывать одно очень важное правило: *все, что следует за словом* **PRINT**, *должно быть заключено в кавычки.*

Например,

если ты напишешь **PRINT «Я ЛЮБЛЮ ШОКОЛАД»**

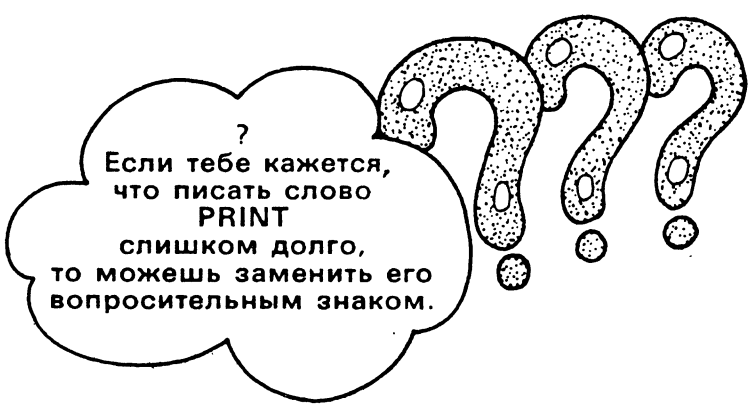
на экране появится **Я ЛЮБЛЮ ШОКОЛАД**

НО

если ты напишешь **Я ЛЮБЛЮ ШОКОЛАД**

компьютер ответит ∅

Вообще-то компьютер обычно бывает очень послушным, но он *не умеет думать*, он только в точности выполняет твои указания. Поэтому их нужно выражать очень ясно. На самом деле именно ты – настоящий мозг компьютера.



Поиграем с инструкцией PRINT

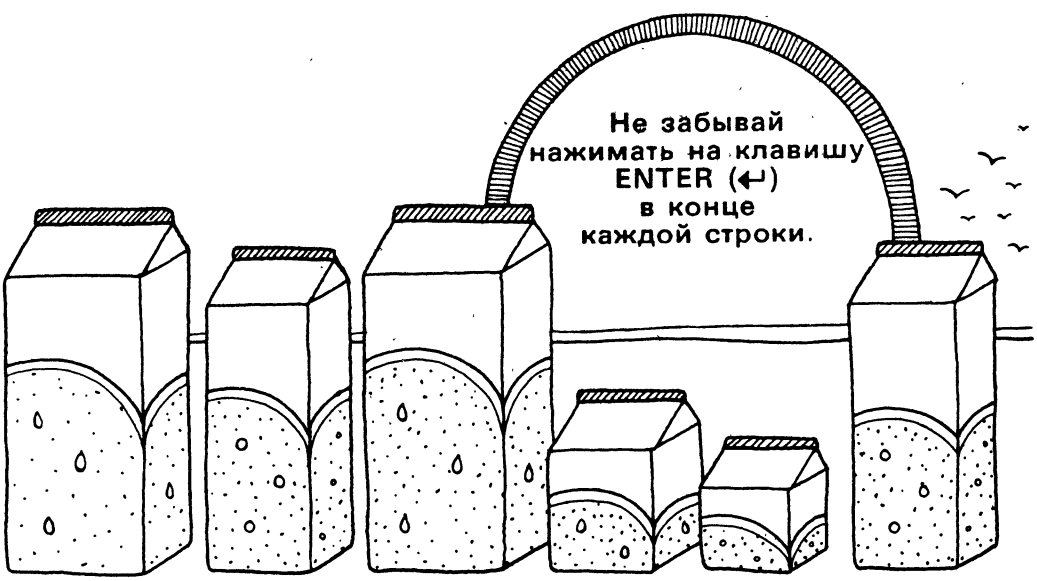
1. Очисти экран.

2. Напиши: PRINT «ДАЙТЕ МНЕ, ПОЖАЛУЙСТА, ПОЛКИЛО МОЛОКА». ? «ЭТО НЕВОЗМОЖНО. МОЛОКО НЕ ПРОДАЕТСЯ НА ВЕС». PRINT ТОГДА ДАЙТЕ МНЕ ПОЛМЕТРА МОЛОКА.

Выполнил ли компьютер твои указания?

3. Исправь третью инструкцию так, чтобы компьютер отобразил ее на экране.

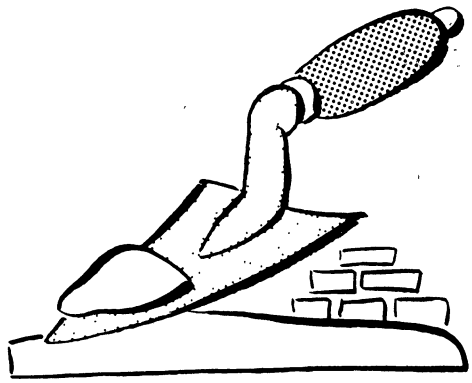
4. Предложи компьютеру изобразить:
твое имя;
забавную историю;
название твоей любимой книги.



ПРОГРАММИРОВАНИЕ

Чтобы получать от пользования компьютером настоящее удовольствие, ты должен научиться *программировать,* т. е. составлять программы.

Программа – это последовательность инструкций, написанных на таком языке, который понятен компьютеру.



Чтобы компьютер знал, в каком порядке ему надо выполнять твои инструкции, ты должен каждой из них дать свой номер.

Можно, конечно, пронумеровать их по порядку: 1, 2, 3 и т. д., но будет лучше, если ты их пронумеруешь десятками: 10, 20, 30 и т. д. Так будет удобнее в том случае, если ты пропустил какую-нибудь инструкцию и тебе придется ее потом вставить.

10
20
30
40
50

Если ты написал, например,

```
10 PRINT «САША И МАША»
20 PRINT «КУПАТЬСЯ.»
30 END
```

то тебе легко вставить недостающую строку:

```
15 PRINT «ПОШЛИ НА РЕЧКУ»
```

Если бы твои инструкции были пронумерованы 1, 2, 3, то тебе не удалось бы так легко исправить ошибку, потому что нет такого целого числа, которое можно было бы вставить между цифрами 1 и 2.

RUN

Компьютер не начнет выполнять программу до тех пор, пока ты не прикажешь ему начать. Значит, чтобы программа начала работать, ты должен дать команду **RUN**.

Как только компьютер увидит это слово, он начнет читать твою программу по порядку возрастания номеров инструкций, а потом будет выполнять то, что ты от него потребовал. Если ты расположил инструкции не по порядку номеров, то компьютер сам расставит их по нужным местам. Можешь сам убедиться в том, что он хорошо справляется с этой задачей. Напиши следующую программу:

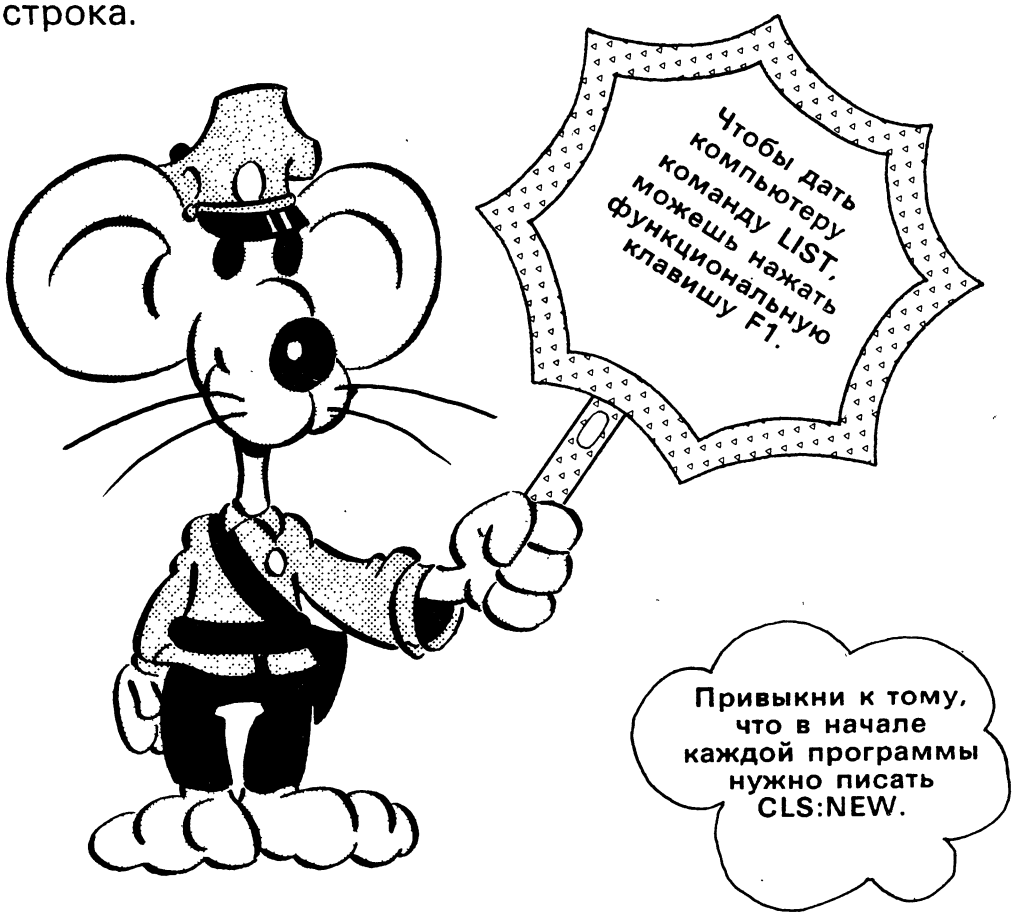
```
CLS:NEW
10 PRINT «ЗДРАВСТВУЙ»
20 PRINT «ПРОГРАММЫШКА»
30 END
15 PRINT «МЕНЯ ЗОВУТ»
```

Прикажи компьютеру начать выполнять программу (**RUN**), а затем напиши команду **LIST**, чтобы убедиться, расставлены ли инструкции в нужном порядке.

NEW и LIST

Выполняя упражнение, описанное на предыдущей странице, ты использовал два новых слова: **NEW** и **LIST**. Это – *команды*, т. е. такие инструкции, которые даются компьютеру в виде простейшего указания. По ним компьютер может выполнять действия над программами: **NEW** позволяет стирать программы, находящиеся в памяти, а **LIST** заставляет отображать их на экране.

Ты можешь использовать команду **LIST** для отображения какой-нибудь одной строки. Для этого достаточно указать ее номер. Так, например, по команде **LIST 10** будет отображена на экране только десятая строка.



Посмотрим, как ты понял.

1. Напиши программу, по которой компьютер изобразил бы на экране следующий текст:

ШЕЛ САША
ПО ШОССЕ
И СОСАЛ
СУШКУ.

2. А теперь вот такую:

```
CLS:NEW
10 ?«ПОНЕДЕЛЬНИК»
20 ?«ВТОРНИК»
30 ?«ЧЕТВЕРГ»
40 ?«ПЯТНИЦА»
50 ?«ВОСКРЕСЕНЬЕ»
60 END
```

Какие дни я забыла написать? Правильно, *среда* и *суббота*. Теперь ты сумеешь самостоятельно исправить мою ошибку?

3. Посмотри внимательно на следующую программу, а затем придумай свой собственный рисунок и напиши для него свою программу.

```
CLS:NEW
10 ?"          * * * *          "
20 ?"     * * *        * * *     "
30 ?"* * *    0        0    * * *"
40 ?"* * *                   * * *"
50 ?"* * *    ( - - - - )    * * *"
60 ?"     * * * * * * * * * *     "
70 END
```

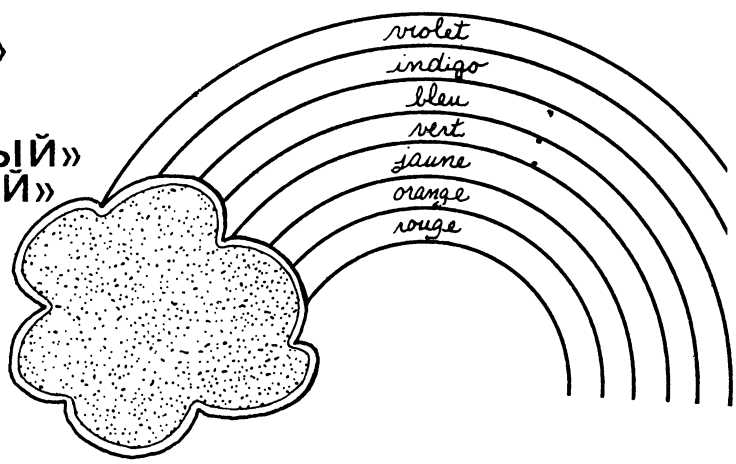
Отладка программы

Часто бывает так, что во время составления программы в нее вкрадываются ошибки. К счастью, большинство из них легко поддается исправлению.

1. Если ты заметил ошибку до того, как нажал на клавишу **ENTER** (↵), то ее легко исправить с помощью курсора.
2. Если ты уже нажал на **ENTER**, то тебе придется переписать эту строку заново. Если хочешь стереть строку, достаточно написать ее номер и нажать на клавишу **ENTER**.

Давай рассмотрим пример. Напиши эту программу, а потом поставь ее на выполнение.

```
CLS:NEW
10 ? «ФИОЛЕТОВЫЙ»
20 ? «СИНИЙ»
30 ? «ГОЛУБОЙ»
40 ? «ЗЕЛЕНЫЙ»
50 ? «ЖЕЛТЫЙ»
60 ? «ЧЕРНЫЙ»
70 ? «ОРАНЖЕВЫЙ»
80 ? «ВИШНЕВЫЙ»
90 END
```



Занятная радуга, не правда ли? А не кажется ли тебе, что один из цветов здесь лишний? (Если ты ответил «черный», то ты молодец!) Чтобы стереть лишнюю строку, напиши ее номер (**60**) и нажми клавишу **ENTER**. Не забудь также исправить строку 80, которая должна выглядеть так: **80 ? «КРАСНЫЙ»**

Нажми на клавишу **ENTER**, а теперь проверь, верно ли компьютер исправил все ошибки (**LIST**).

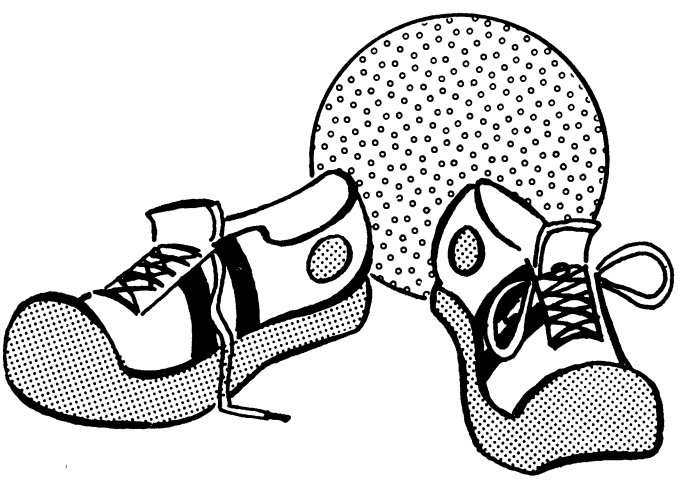
Напиши следующую программу на своем экране, а затем исправь ее, заменяя, стирая или добавляя строки:

```
1. CLS:NEW
  10 ? «ФУТБОЛЬНЫЙ МЯЧ И БУТСЫ»
  20 ? «ВЕЛОСИПЕДНЫЙ НАСОС И КАМЕРА»
  30 ? «ТЕННИСНЫЕ МЯЧИ И РАКЕТКА»
  40 ? «ПИРОГИ С ВАРЕНЬЕМ»
  50 ? «ТРЕНИРОВОЧНЫЙ КОСТЮМ»
  60 END
```

В этом списке не должно остаться ничего, кроме спортивных принадлежностей.

```
2. CLS:NEW
  10 ? «КУПАТЬСЯ»
  20 ? «ХОДИТЬ НА РЫБАЛКУ»
  30 ? «ЛАКОМИТЬСЯ МОРОЖЕНЫМ»
  40 ? «РАБОТАТЬ»
  50 ? «ХОДИТЬ В КИНО»
  60 ? «УХАЖИВАТЬ ЗА ЦВЕТАМИ»
  70 ? «ИГРАТЬ В ШПИОНОВ»
  80 ? «НАВОДИТЬ ПОРЯДОК В ШКАФУ»
  90 ? «ИГРАТЬ В ФУТБОЛ»
  100 END
```

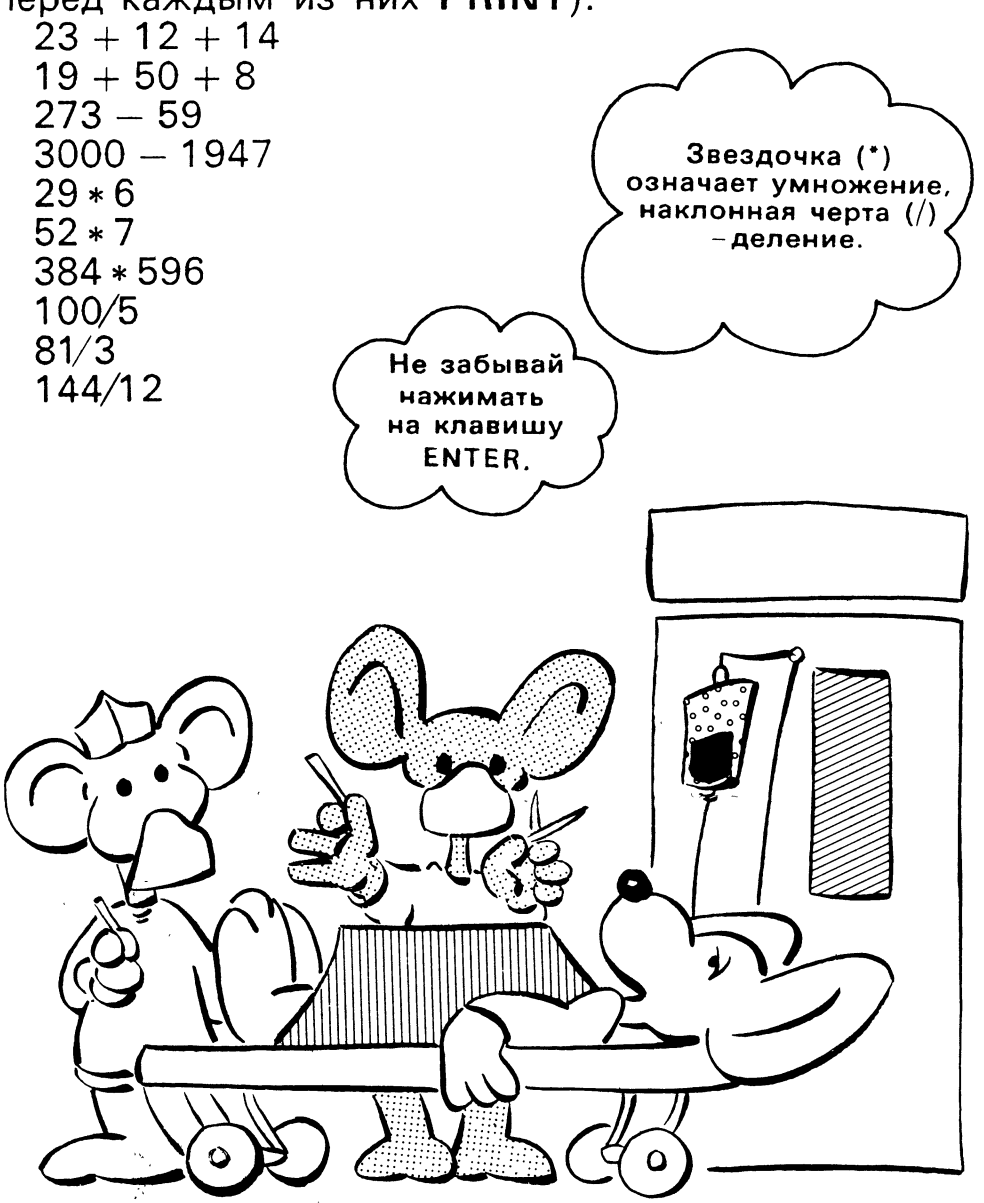
Как, по-твоему, годится этот список в качестве программы на каникулы? Не можешь ли ты его улучшить?



АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

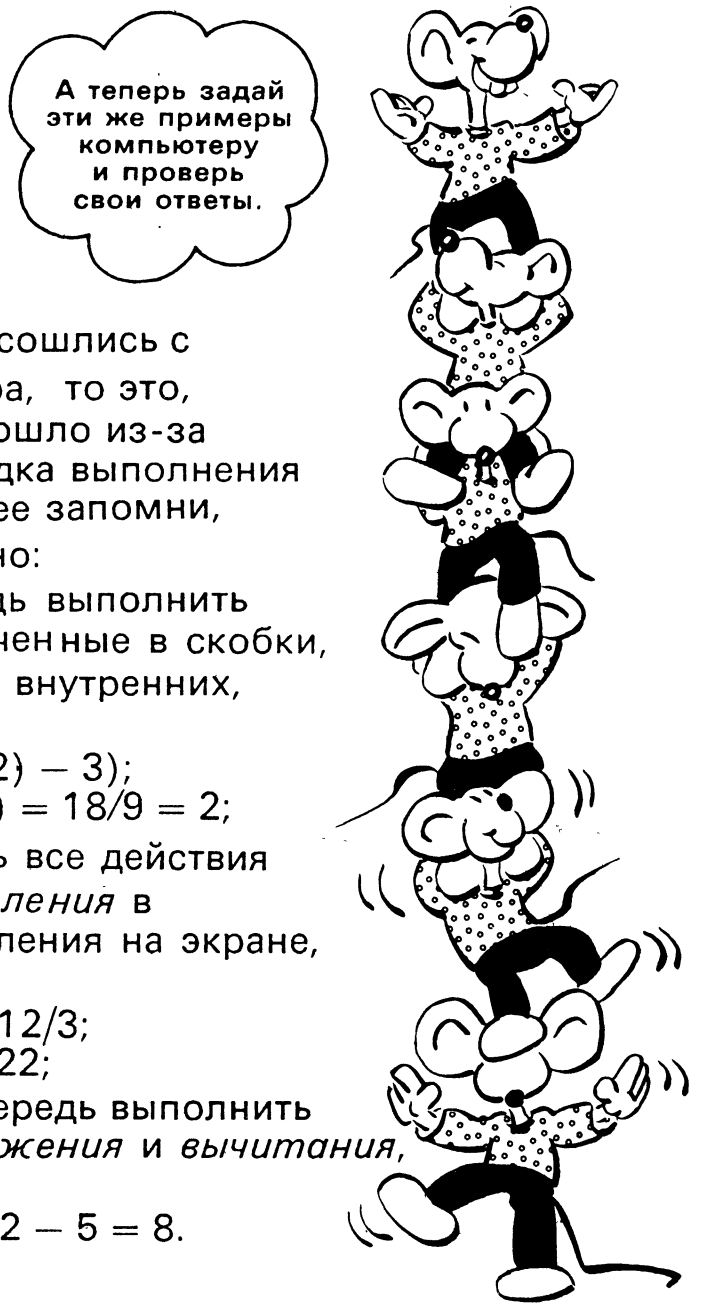
Ты, конечно, знаешь о том, что твой компьютер может выполнять арифметические действия. Надо только правильно попросить его об этом. Дай, например, ему следующие задания (не забудь написать перед каждым из них **PRINT**):

```
23 + 12 + 14
19 + 50 + 8
273 – 59
3000 – 1947
29 * 6
52 * 7
384 * 596
100/5
81/3
144/12
```



Интересно, а ты сам умеешь считать? Ну-ка, реши следующие примеры:

$3 + 7 - 4 + 1 =$
$4 * 7 =$
$6/3 + 2 =$
$6 - 4/2 =$
$4 * 2 + 6/3 + 8 =$
$2 * 2 + 3 =$
$2 * (2 + 3) =$
$2 + 9/3 =$
$12/(4 + 2) =$
$12/4 - 2 =$



Если твои ответы не сошлись с ответами компьютера, то это, скорее всего, произошло из-за несоблюдения порядка выполнения действий. На будущее запомни, что надо обязательно:

в первую очередь выполнить действия, заключенные в скобки, начиная с самых внутренних, например

$18/(6 + (4 + 2) - 3);$
$18/(6 + 6 - 3) = 18/9 = 2;$

затем выполнить все действия *умножения* и *деления* в порядке их появления на экране, например

$3*4 + 2*3 + 12/3;$
$12 + 6 + 4 = 22;$

в последнюю очередь выполнить все действия *сложения* и *вычитания*, например

$12 + 3 - 4 + 2 - 5 = 8.$

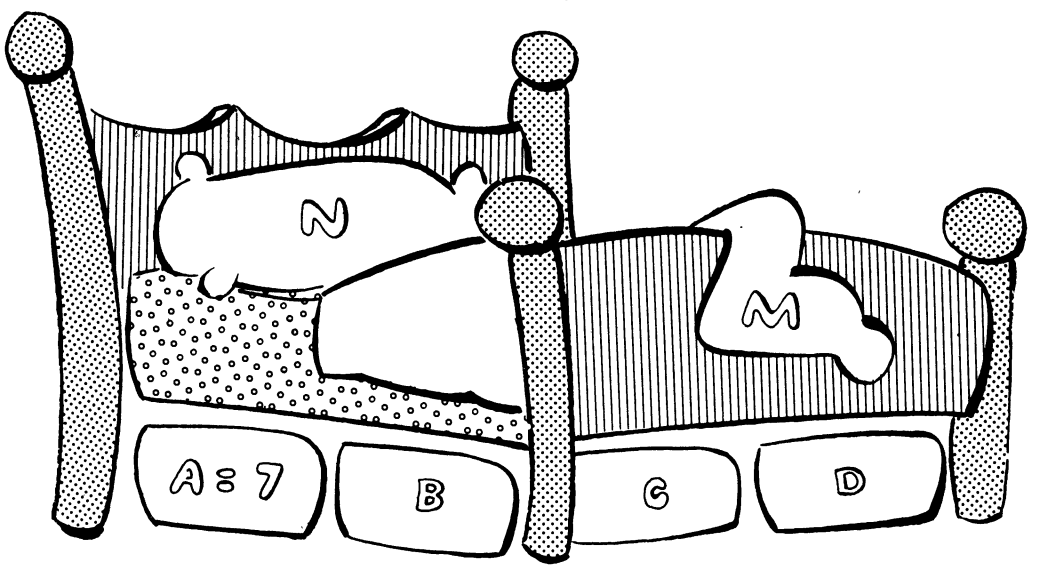
ПЕРЕМЕННЫЕ

Когда ты хочешь что-то сохранить, куда ты это «что-то» деваешь? Под подушку? В носок? В ящик стола? Конечно, можно и так, но что ты станешь делать, если это «что-то» не вещь, которую можно подержать в руках? Разве у тебя есть ящички в голове для запоминания? А у компьютера они есть, и их очень много. Они называются **переменные**.

Ты можешь доверить компьютеру сохранить все, что хочешь. Только имей в виду, что информацию потом придется снова в нем разыскивать, поэтому надо как-то назвать тот ящичек, в котором она будет храниться. Давай рассмотрим пример. Напиши эту программу, а потом выполни ее.

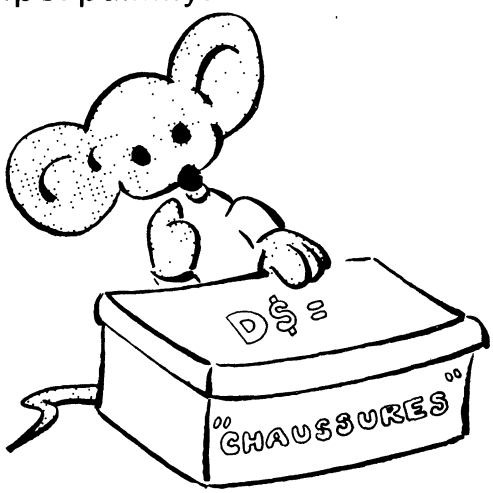
```
CLS:NEW
10 A=7
20 PRINT A
30 END
```

В этом случае цифра 7 хранится в том ящичке, который ты назвал буквой **A**. И теперь до тех пор, пока ты не сотрешь информацию, хранящуюся в ящичке под названием **A**, ты можешь сколько угодно раз пользоваться этой информацией, заставляя компьютер отображать содержимое переменной **A**.



В БЕЙСИКЕ имя переменной должно обязательно начинаться с буквы (**A, B, C, ...**). За этой первой буквой может идти буква, или цифра (**AB, A1, ...**), и символ **$**.
Посмотри внимательно на программу.

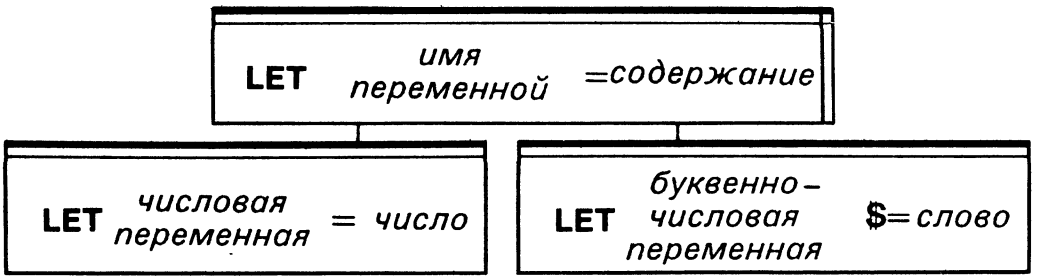
```
CLS:NEW
10 A$=«В КОРОБКЕ»
20 B=1
30 C$=«ЛЕЖАТ»
40 D$=«БОТИНКИ»
50 E=2
60 E$=«У МЕНЯ»
70 PRINT E$, E, D$, C$, B, A$
80 END
```



Обрати внимание: ящички, в которых находится слово, имеют в своем имени символ **$**, а само слово, находящееся в ящичке, заключено в *кавычки.* Постарайся запомнить, это очень важно.

LET

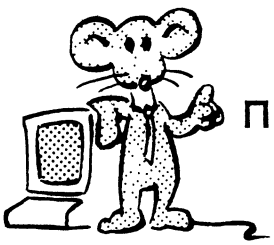
Для размещения информации в памяти компьютера можно использовать инструкцию **LET**. Инструкция всегда одна и та же, изменяется только имя переменной и ее содержание.



Примеры:

```
10 LET A=5
20 LET B=3
30 LET C=A * B
```

```
50 LET A$= «УМНОЖИТЬ НА»
60 LET B$= «РАВНО»
70 LET C$= «МОЛОДЕЦ!»
```



Поиграем с переменными

1. Догадайся, что получится в результате выполнения следующей программы:

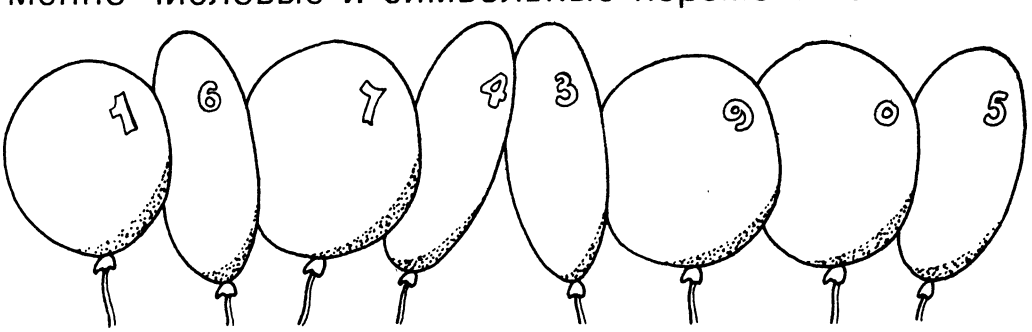
```
CLS:NEW
10 A=3
20 B=4
30 C=5
40 PRINT A + B
50 PRINT B + C
60 PRINT A * C
70 END
```

Предложи компьютеру выполнить эту программу.

2. Напиши программу, по которой компьютер умножит число 6 на 8 и поместит результат в ящичек под номером 2.

3. Попроси компьютер написать: «МОЕГО ДРУГА ЗОВУТ ПРОГРАММЫШКА». Если хочешь, можешь использовать несколько переменных.

4. Напиши программу, в которой участвуют одновременно числовые и символьные переменные.



INPUT

Что произойдет, если ты попросишь компьютер выполнить следующую программу?

```
CLS:NEW
10 ? «ЗДРАВСТВУЙ! КАК ТЕБЯ ЗОВУТ?»
20 INPUT R$
30 ? «ОЧЕНЬ ПРИЯТНО!»
40 END
```

После того как будет отображена строка 10?, выполнение программы приостановится. Действительно, инструкция **INPUT** требует, чтобы компьютер дождался от тебя ответа (**R $**) перед тем, как продолжить выполнение программы. Старайся отвечать побыстрее.

Ты заметил, что компьютер изображает вопросительный знак каждый раз после того, как прочтет инструкцию **INPUT**?

Чтобы лучше понять роль инструкции **INPUT**, проверь следующую программу, а затем внеси в нее предлагаемые изменения:

```
CLS:NEW
10 ?«ЗДРАВСТВУЙ! КАК ТЕБЯ ЗОВУТ?»
20 INPUT X$
30 ?«ОЧЕНЬ ПРИЯТНО! □";X$;"□ КАК
                              ПОЖИВАЕШЬ?»
40 INPUT Y$
50 ?Y$;"?□Я ТОЖЕ»
60 END
```

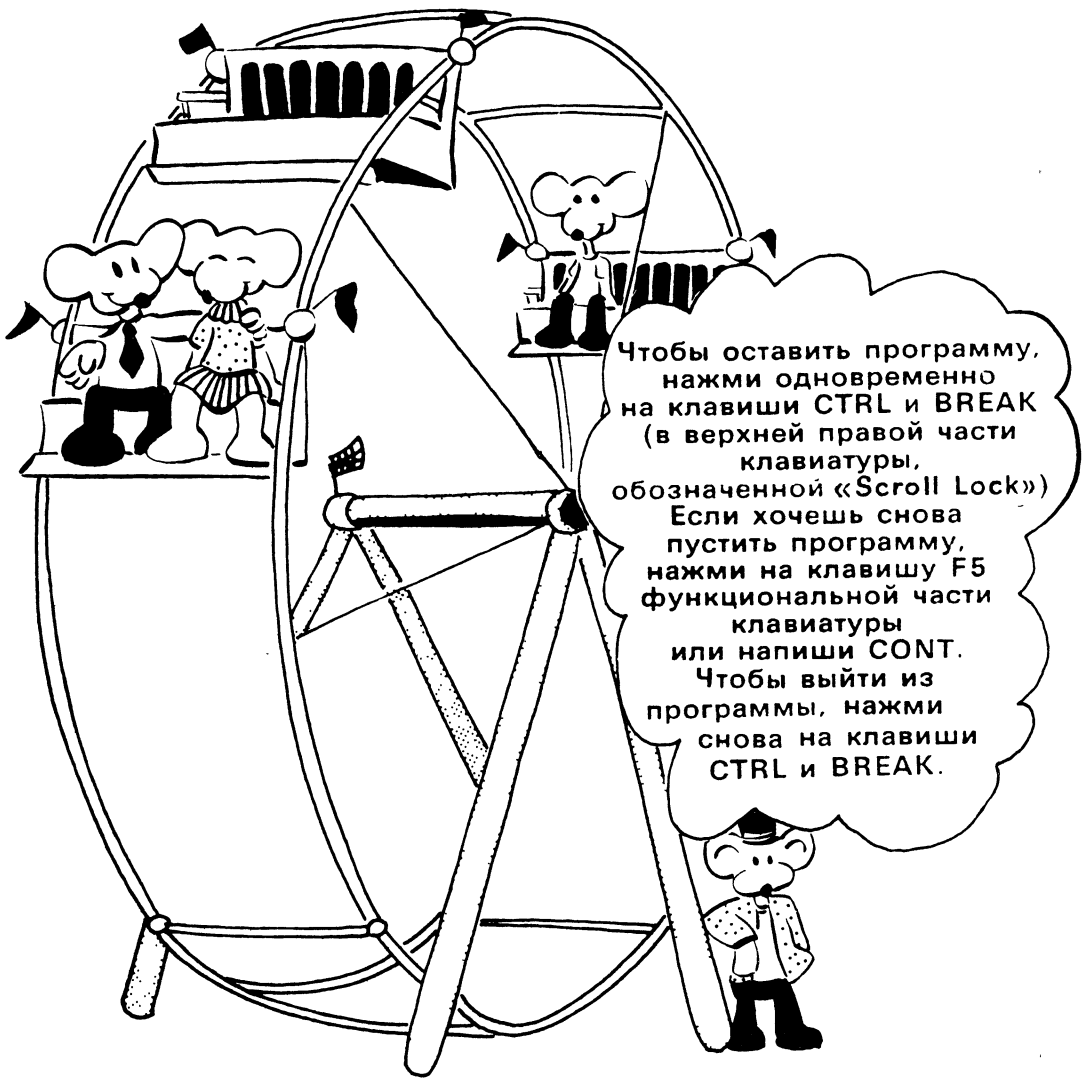
замени вопрос в строке 10,
измени наименования переменных в строках 20 и 40,
сотри пробелы (□).

А теперь придумай такой диалог, в котором компьютер сам задавал бы тебе вопросы. Для продления разговора используй несколько раз инструкцию **INPUT**.

ЦИКЛ

Посмотри внимательно, что произойдет, если ты дашь компьютеру следующее задание:

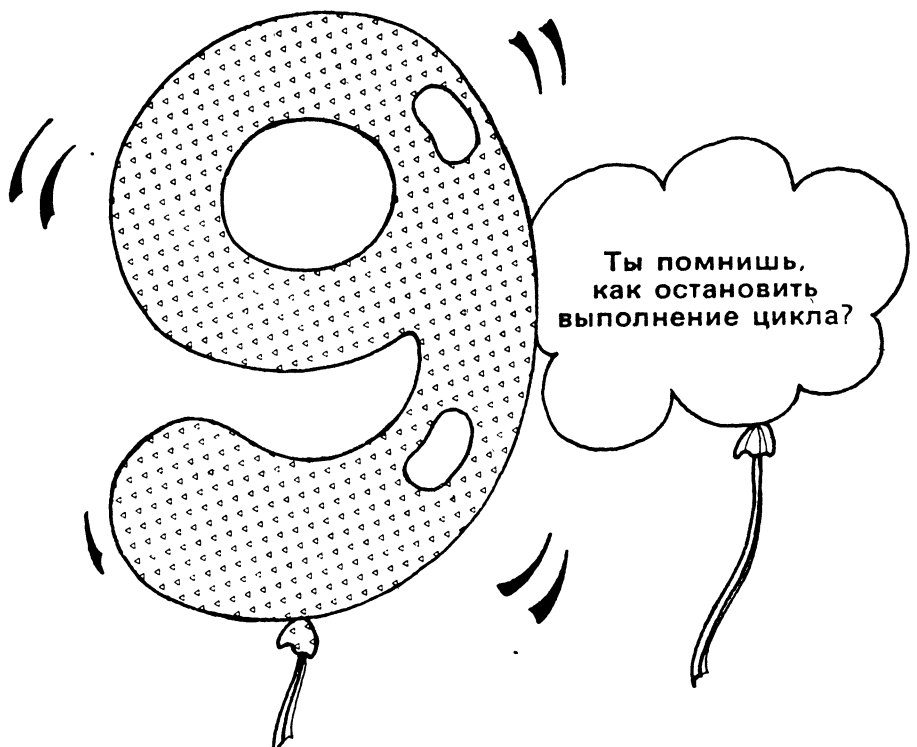
```
CLS:NEW
10 ?«А КОЛЕСО ВСЕ КРУТИТСЯ»
20 ?«И КРУТИТСЯ»
30 GOTO 20
```



GOTO

Инструкция **GOTO** позволяет изменить ход выполнения программы. Действительно, если ты сотрешь строку 30 в программе, написанной на предыдущей странице, компьютер остановится, выполнив все команды один раз.
Возьмем другой пример.

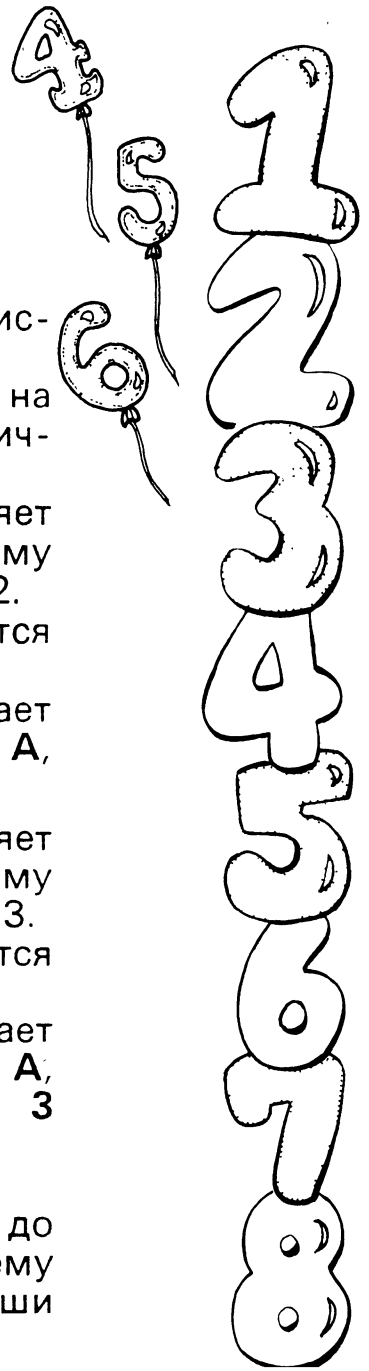
```
CLS:NEW
10 ?«ПОСМОТРИ, КАК Я УМЕЮ УМНОЖАТЬ
                                НА 9»
20 ?«ВЫБЕРИ ЧИСЛО»
30 INPUT X
40 ?X;«□ УМНОЖИТЬ НА 9 РАВНО□ »;X*9
50 ?«НУ КАК, УБЕДИЛСЯ?»
60 GOTO 20
```



Компьютер очень хорошо считает. Чтобы убедиться в этом, замени цифру 9 на любую другую.
Если хочешь, можешь изменить диалог.

Компьютер может считать как бы сам по себе. Тебе достаточно только попросить его об этом.

```
CLS:NEW
10 A=1
20 ? A
30 A=A+1
40 GOTO 20
```



Вот что при этом происходит.

Строка 10 – компьютер помещает число 1 в ящичек **A**.

Строка 20 – компьютер отображает на экране содержимое ящичка **A**, т. е. число **1**.

Строка 30 – компьютер прибавляет число 1 к содержимому ящичка **A**, т. е. 1 + 1 = 2.

Строка 40 – компьютер возвращается к строке 20.

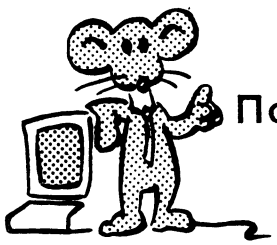
Строка 20 – компьютер отображает содержимое ящичка **A**, т. е. число **2**.

Строка 30 – компьютер прибавляет число 1 к содержимому ящичка **A**, т. е. 2 + 1 = 3.

Строка 40 – компьютер возвращается к строке 20.

Строка 20 – компьютер отображает содержимое ящичка **A**, которое сейчас равно **3** и т. д.

Компьютер будет продолжать счет до тех пор, пока ты не предложишь ему остановиться, нажав на клавиши (**CTRL**/**BREAK**).



Поиграем с циклами

1. Набери снова программу, написанную на предыдущей странице, заменив в ней строку 30:

30 A=A+5

Внимательно посмотри, что произойдет. Затем измени программу так, чтобы компьютер считал:
десятками (от 10 через 10),
двадцатками (от 20 через 20),
полусотнями (от 50 через 50).

2. Напиши программу, в которой компьютер использовал бы твое имя в цикле.

3. Попробуй нарисовать что-нибудь, используя инструкцию **PRINT** и цикл.

АВТОМАТИЧЕСКАЯ НУМЕРАЦИЯ

Если тебе кажется, что писать самому номера в начале каждой строки программы – дело скучное, ты можешь поручить это компьютеру. Он справится, надо только его правильно попросить.

Чтобы он хорошо тебя понял, напиши слово **AUTO** и нажми на клавишу **ENTER** (↵). Ты увидишь, как на экране появится число **10**.

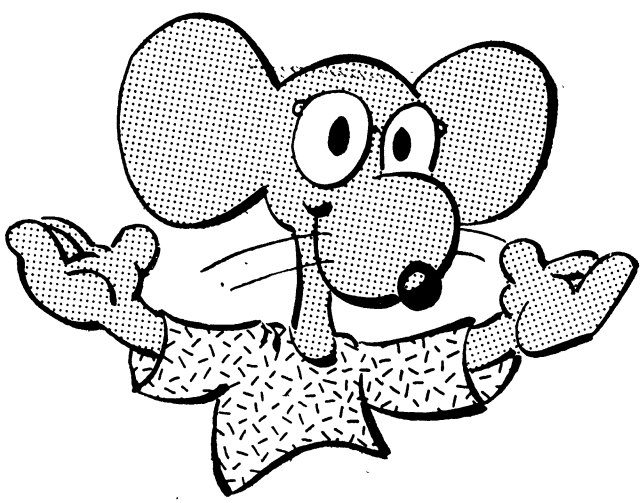
Теперь добавь, например, такую инструкцию:
?«ПРОГРАММЫШКА ЗАБРАЛАСЬ КО МНЕ В ПОРТФЕЛЬ» и снова нажми на клавишу **ENTER**.
Видишь, как просто?

Компьютер прекратит считать, когда ты нажмешь на клавиши **CTRL** и **BREAK**.

Если ты хочешь, чтобы компьютер нумеровал строки от 5 по 5, напиши **AUTO, 5**

от 2 по 2, напиши **AUTO, 2**

от 20 по 20, напиши
AUTO, 20
и т. д.

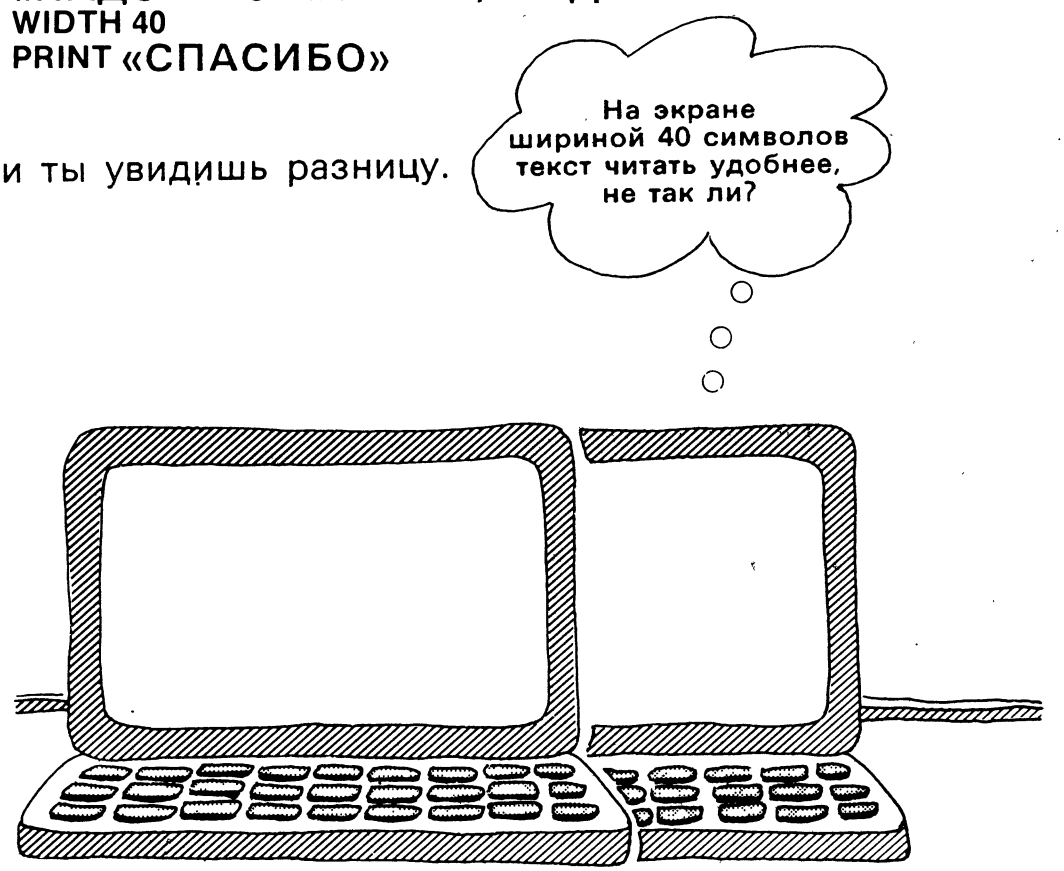


РАЗМЕР ЭКРАНА

Ты, наверное, пока еще не успел заметить, что компьютер может размещать на одной строке до 80 символов. Как только ты превысишь это число, курсор перепрыгнет на следующую строку. Если у тебя хватит терпения, можешь сам убедиться в этом, написав 81 раз какую-нибудь букву.
Мне кажется, что 80 символов в строке – это многовато. Я тебе советую уменьшить это число вдвое. Перепиши следующие команды:

```
«НАДО ПРОВЕТРИТЬ, НАДО ПРОВЕТРИТЬ»
WIDTH 40
PRINT «СПАСИБО»
```

и ты увидишь разницу.



ПУНКТУАЦИЯ

Перепиши внимательно эти две программы и выполни их.

```
CLS:NEW
10 ? «Я ТЕБЯ ЛЮБЛЮ»
20 GOTO 10
```

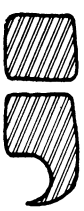
```
CLS:NEW
10 «Я ТЕБЯ ЛЮБЛЮ»;
20 GOTO 10
```

Ты заметил разницу между ними?
А теперь попробуй еще две:

```
CLS:NEW
10 ? 1 2 3 4 5
20 GOTO 10
```

```
CLS:NEW
10 ? 1, 2, 3, 4, 5
20 GOTO 10
```

Посмотри, какую важную роль играют знаки препинания – *запятая* и *точка с запятой.*



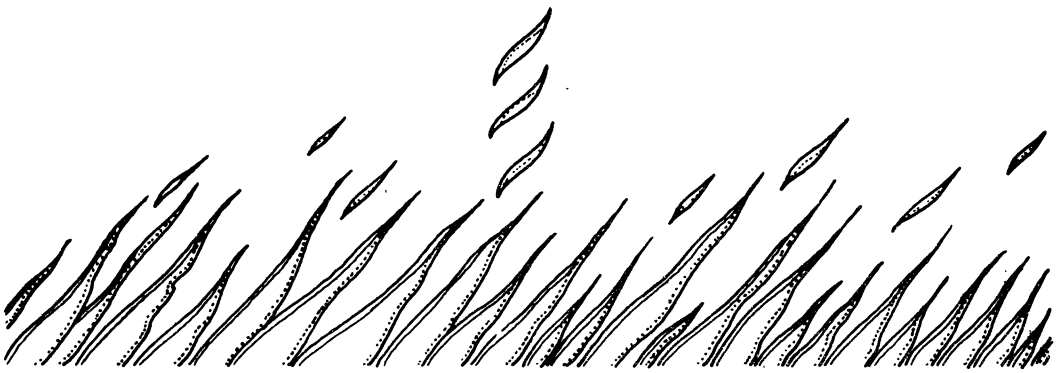
Точка с запятой заставляет компьютер не менять строку. Вот почему он пишет подряд много раз Я ТЕБЯ ЛЮБЛЮ на одной строке.

Запятая советует компьютеру перейти в следующую зону отображения или следующий столбец. Поскольку цифр всего пять, он отображает 1, 2, 3, 4 и 5 на одной строке. Если бы компьютер получил шестую цифру, ему надо было бы отображать ее на следующей строке в первом столбце.

Посмотрим, как ты понял это.

1. Заполни экран шириной 40 символов словами **«НА ПОЖАР!»**

НА ПОЖАР! НА ПОЖАР! НА ПОЖАР!
НА ПОЖАР! НА ПОЖАР! НА ПОЖАР!
НА ПОЖАР! НА ПОЖАР! НА ПОЖАР!



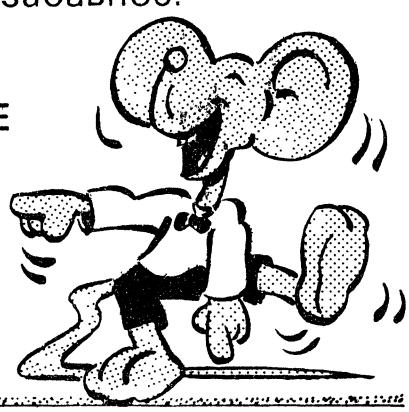
2. Изобрази числа 10, 20, 30, 40 и 50, используя пять столбцов на твоем полном экране шириной 80 символов.

3. Заполни экран своим собственным именем, используя точку с запятой, а потом запятую. Проделай это упражнение еще раз с экраном шириной 40 символов.

4. Заполни экран названием твоего любимого пирожного.

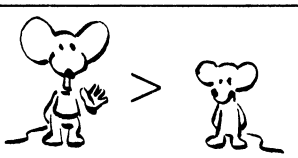
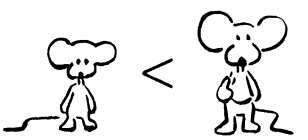
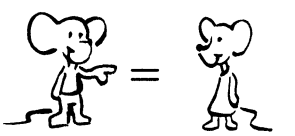
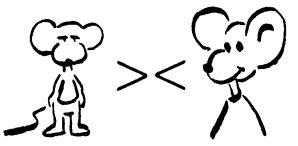
5. Взяв в качестве образца следующую программу, изобрази на экране что-нибудь забавное.

```
CLS:NEW
10 ?«ПРИДУМАЙ СМЕШНОЕ
     СЛОВЕЧКО»
20 INPUT M$
30 ? M$
40 GOTO 30
```



СРАВНЕНИЯ

В жизни часто приходится делать всевозможные сравнения, например:

Я *больше* Малыша.	
Малыш *меньше* меня.	
Рост братца *равен* моему росту.	
Рост моего приятеля *отличается* от моего.	

Если ты хорошо понял смысл знаков $>$, $<$, $=$ и $<>$, тебе будет легко найти ошибки в следующих примерах. Какие из них неверные?

а) $8 <> 6$
б) $6 + 3 = 4 + 5$
в) $3 - 1 > 1$
г) 30 минут = 1/2 часа
д) 50 минут $<$ 1 часа
е) $2 * 8 = 15$
ж) $12 <>$ 1 дюжина
з) $12/4 < 3$

Ответ: е, ж и з

IF ... THEN

Компьютер может выполнять сравнение также с помощью оператора **IF ... THEN**. Если твоя программа не хочет выходить из цикла, то тебе достаточно позвать на помощь эту инструкцию, и она с удовольствием поможет тебе остановить непослушную программу.

Напиши вот эту небольшую программу, и ты сам убедишься в этом.

```
CLS:NEW
10 A=1
20 ? A
30 A=A+1
40 GOTO 20
50 END
```

А можешь получить то же самое другим способом: замени строку 40 на такую:

40 IF A > 20 THEN GOTO 50

и при этом добавь еще одну:

45 GOTO 20

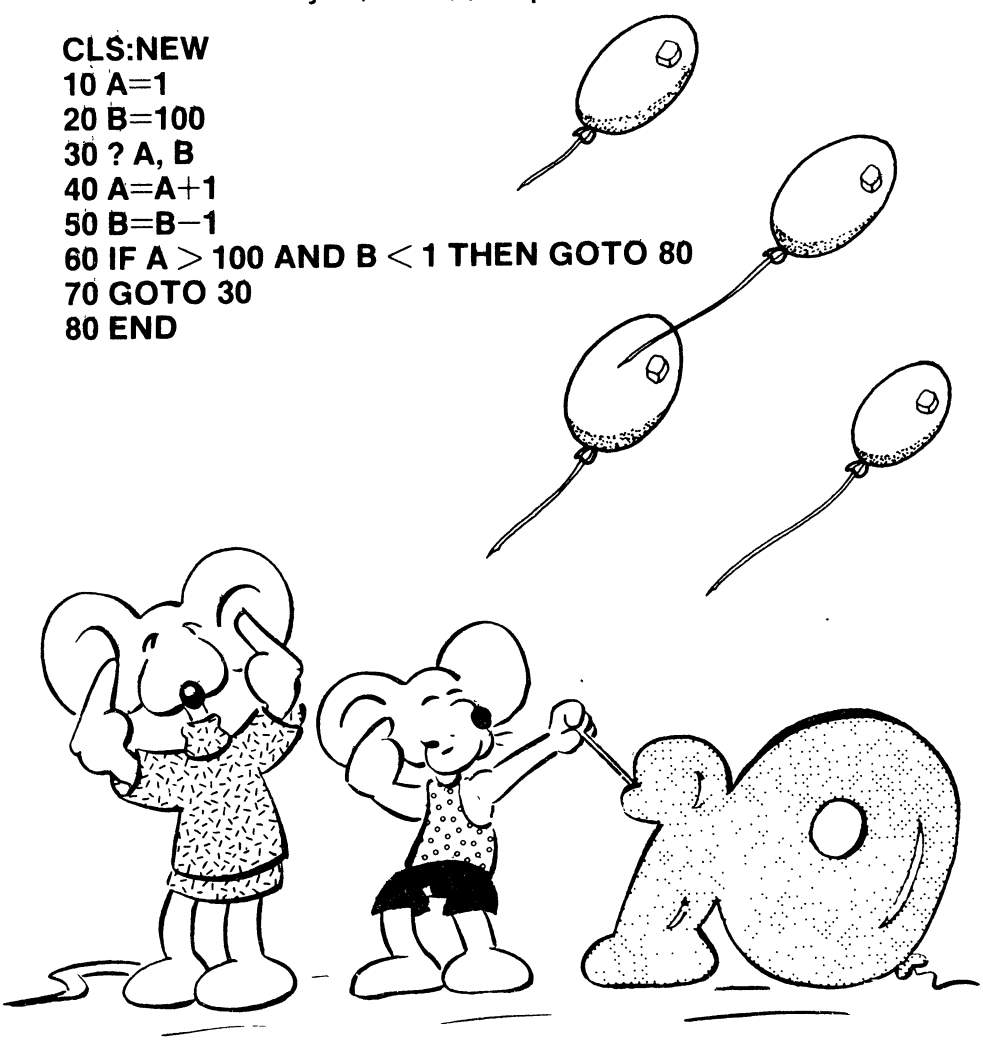
В обоих случаях компьютер будет циклически выполнять запланированные операции до тех пор, пока **A** меньше чем 21.

Пока шарик еще не лопнул, я полагаю, было бы хорошо, чтобы ты на практике опробовал то, что тебе только что стало известно.

1. Предложи компьютеру:
   считать от 5 до 50 через 5, а результаты отображать в столбцах на экране шириной 40 символов;
   считать «задом наперед» от 100 до 0, а результат отображать на экране шириной 80 символов.

2. Взяв за образец следующую программу, составь свою собственную, найдя оригинальные комбинации.

```
CLS:NEW
10 A=1
20 B=100
30 ? A, B
40 A=A+1
50 B=B-1
60 IF A > 100 AND B < 1 THEN GOTO 80
70 GOTO 30
80 END
```



3. Напиши следующую программу:

```
CLS:NEW
10 ? «ОТГАДАЙ, КАКОЕ ЧИСЛО
     ЗАДУМАНО!»
20 INPUT A
30 IF A > 7 THEN GOTO 60
40 IF A < 7 THEN GOTO 80
50 IF A = 7 THEN GOTO 100
60 ? «ОЧЕНЬ ЖАЛЬ, НО ЭТО ЧИСЛО
     СЛИШКОМ ВЕЛИКО»
70 GOTO 10
80 ? «ОЧЕНЬ ЖАЛЬ, НО ЭТО ЧИСЛО
     СЛИШКОМ МАЛО»
90 GOTO 10
100 ? «БРАВО! МОЛОДЕЦ!»
110 END
```

Теперь замени строку 50 на такую:

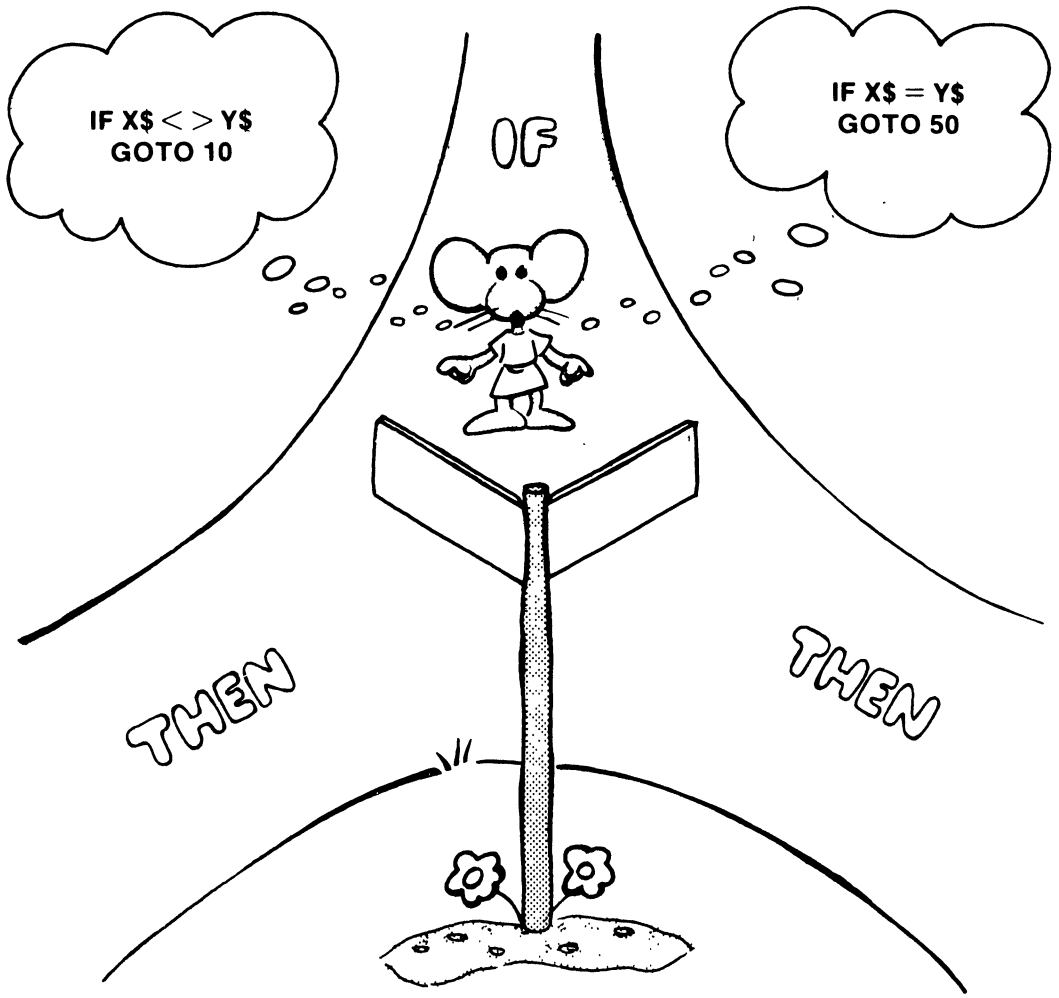
```
50 IF A=50 THEN GOTO 100
```

Скажи, надо ли заменить еще какие-нибудь строки, чтобы программа заработала правильно? Какие?

4. Можешь ли ты изменить программу, начиная с вопроса **3** так, чтобы она стала похожа на эту:

```
CLS:NEW
10 INPUT «ВЫБЕРИ ЧИСЛО»; T
20 IF T < > 9 THEN GOTO 50
30 ? «БРАВО!»
40 GOTO 70
50 ? «НЕТ, НЕ ЭТО»
60 GOTO 10
70 END
```

Ответственный момент: принятие решения




```
CLS:NEW
10 ?«КАК ЗОВУТ ТВОЕГО ГИДА?»
20 INPUT X$
30 IF X$ = «ПРОГРАММЫШКА» THEN 50
40 IF X$ < > «ПРОГРАММЫШКА» THEN 10
50 ?«ПРАВИЛЬНО»
60 END
```

Ты заметил, что в строках 30 и 40 не обязательно писать **GOTO**?

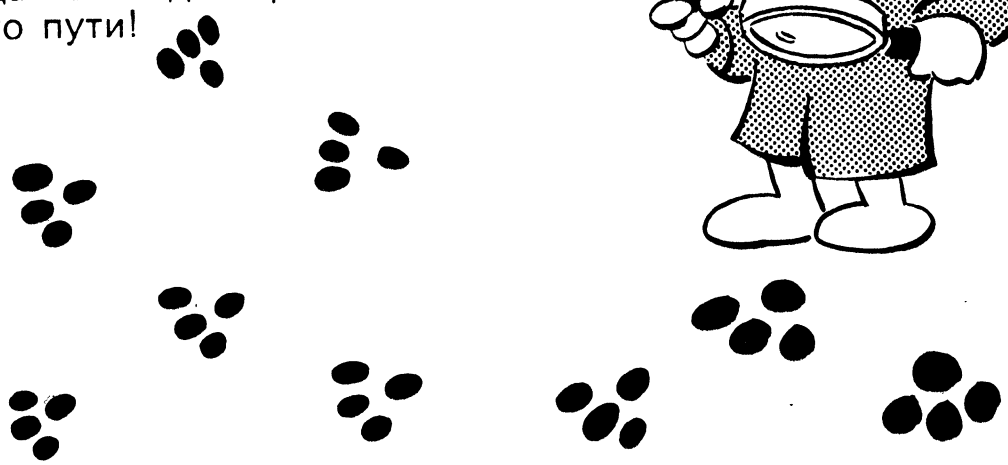
Когда ты просишь компьютер выполнить сравнение и сделать вывод, то на самом деле заставляешь его принимать решение.

Рассмотрим пример:

```
CLS:NEW
10 ? «ЗАДУМАЙ ДВА ЧИСЛА»
20 INPUT X
30 INPUT Y
40 ? «ЧЕМУ РАВНО ПРОИЗВЕДЕНИЕ
50 INPUT Z                    ЭТИХ ЧИСЕЛ?»
60 IF X * Y <> Z THEN 40
70 IF X * Y = Z THEN 80
80 ? «ЗАМЕЧАТЕЛЬНО!»
90 END
```

В этой программе компьютер должен принимать решение в зависимости от того, чему равно число Z (т. е. от значения, которое ты присвоил переменной Z). Если твой ответ правильный, то будет выполняться строка 80, но если ответ неверный, то выполнится переход к строке 40, т. е. вопрос повторится.

Не забывай, что ты — настоящий мозг твоего компьютера. Если ты ошибся, давая ему инструкции, то он сам никогда не найдет правильного пути!



```
CLS:NEW
10 INPUT «КАК ТЕБЯ ЗОВУТ»;N$
20 IF N$ = «УСАТИК» THEN 60
30 ? «ЭТО ЧАСТНЫЙ КЛУБ»
40 ? «КОТОВ НЕ ПРИНИМАЕМ»
50 GOTO 80
60 ? «ДОБРО ПОЖАЛОВАТЬ В КЛУБ
                          ДЕТЕКТИВОВ»
70 ? «ПАРОЛЬ – ПРОГРАММЫШКА»
80 END
```

Ты можешь использовать инструкцию IF ... THEN, чтобы загадывать своим друзьям загадки.

```
CLS:NEW
10 ?«КОГДА КОНЯ ПОКУПАЮТ»
20 INPUT «КАКОЙ ОН БЫВАЕТ?» ;X$
30 Y$ = «МОКРЫЙ»
40 IF X$ = Y$ THEN 70
50 ?«НЕТ, НЕ УГАДАЛ. ПОПРОБУЙ ЕЩЕ РАЗ.»
60 GOTO 10
70 ?«МОЛОДЕЦ, ПРАВИЛЬНО ДОГАДАЛСЯ!»
80 END
```

А теперь попробуй загадать такую загадку:

– Почему птицы осенью улетают на юг?

– Потому, что идти туда пешком слишком далеко!

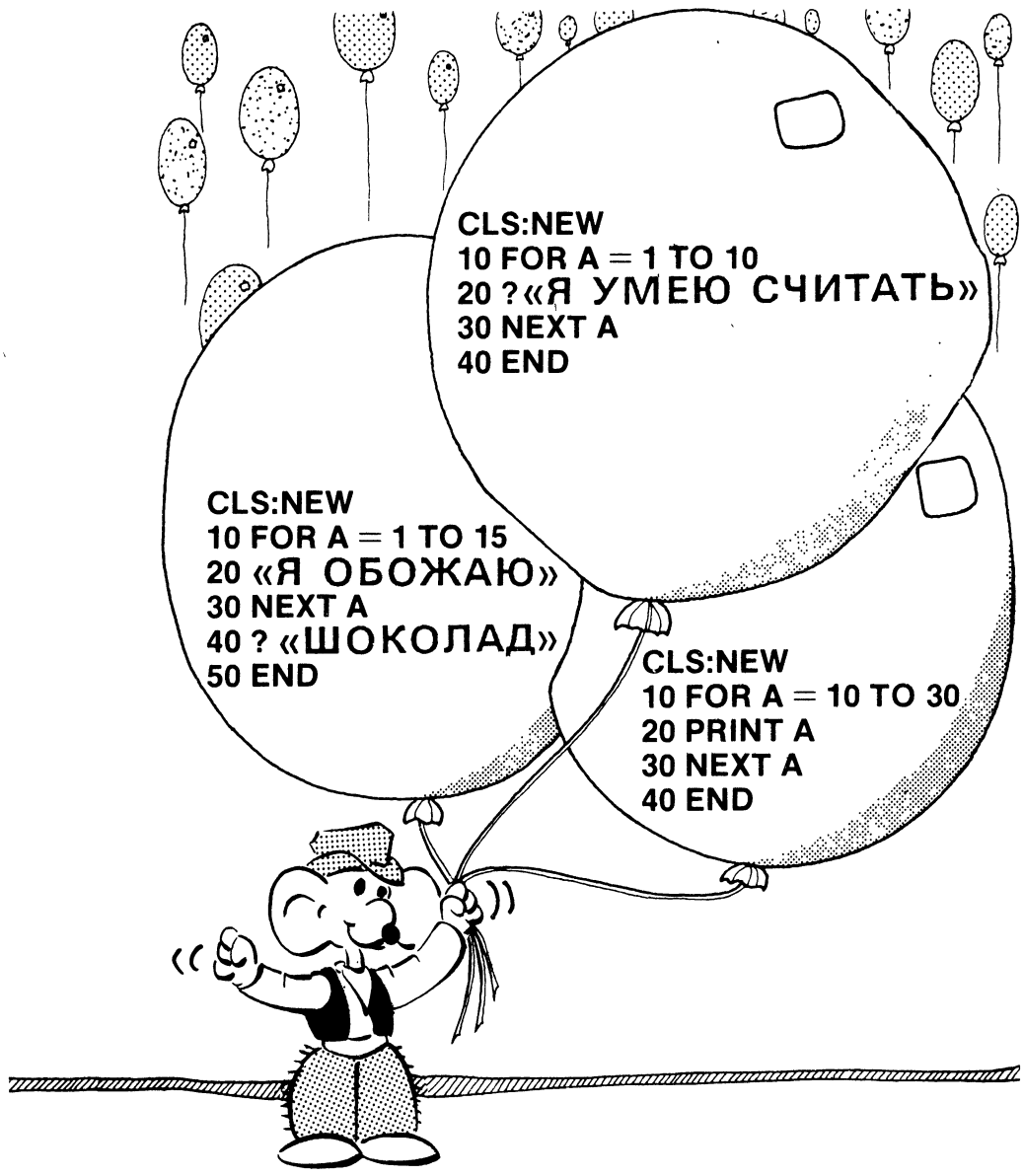
Если ты введешь в компьютер побольше информации, то создастся впечатление, будто он в самом деле поддерживает с тобой разговор как человек.

```
CLS:NEW
10 INPUT «КАКОЕ СЕЙЧАС ВРЕМЯ ГОДА»;S$
20 IF S$ = «ЗИМА» THEN PRINT «ЧУДЕСНО! ПОШЛИ
        КАТАТЬСЯ НА ЛЫЖАХ!»
30 IF S$ = «ВЕСНА!»    THEN PRINT «ДАВАЙ
        ПУСКАТЬ КОРАБЛИКИ ПО ЛУЖАМ»
40 IF S$ =«ЛЕТО» THEN PRINT «ДА ЗДРАВСТВУЮТ
        КАНИКУЛЫ!»
50 IF S$ = «ОСЕНЬ»   THEN PRINT «ПОЙДЕМ
        СОБИРАТЬ БУКЕТ ИЗ ОПАВШИХ
        ЛИСТЬЕВ»
60 END
```

Попробуй составить программу подобного типа на тему о дожде и хорошей погоде или о любимом виде спорта.

FOR/NEXT

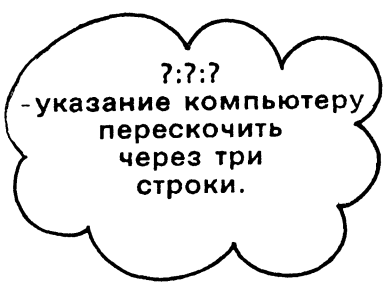
Изучи следующие программы, и тогда увидишь другие способы создания циклов:



Как видишь, если пользоваться инструкцией **FOR/NEXT**, то нет необходимости добавлять инструкцию **IF ... THEN** для остановки цикла.

С помощью инструкции **FOR/NEXT** ты можешь даже заставить рисунок двигаться. Смотри как это делается:

```
CLS:NEW
10 FOR A = 1 TO 20
20 ?:?:?
30 ?"        ( ( ( ( ( ( ( (   ) ) ) ) ) ) )     "
40 ?"          $       -           -     $       "
50 ?"      $           0           0         $   "
60 ?"      $                 =                 $   "
70 ?"      $                                   $   "
80 ?"          $       WWWWW             $       "
90 ?"            - - - - - - - - - - - - -       "
100 ?"             X                   X         "
110 ?"       X                               X   "
120 ?"   X                                     X "
130 ?:?:?
140 NEXT A
150 END
```



Посмотрим, как ты это понял.

1. Придумай сам движущийся рисунок.

2. Измени приведенную ниже программу так, чтобы получилась таблица умножения на восемь любых чисел от 1 до 20.

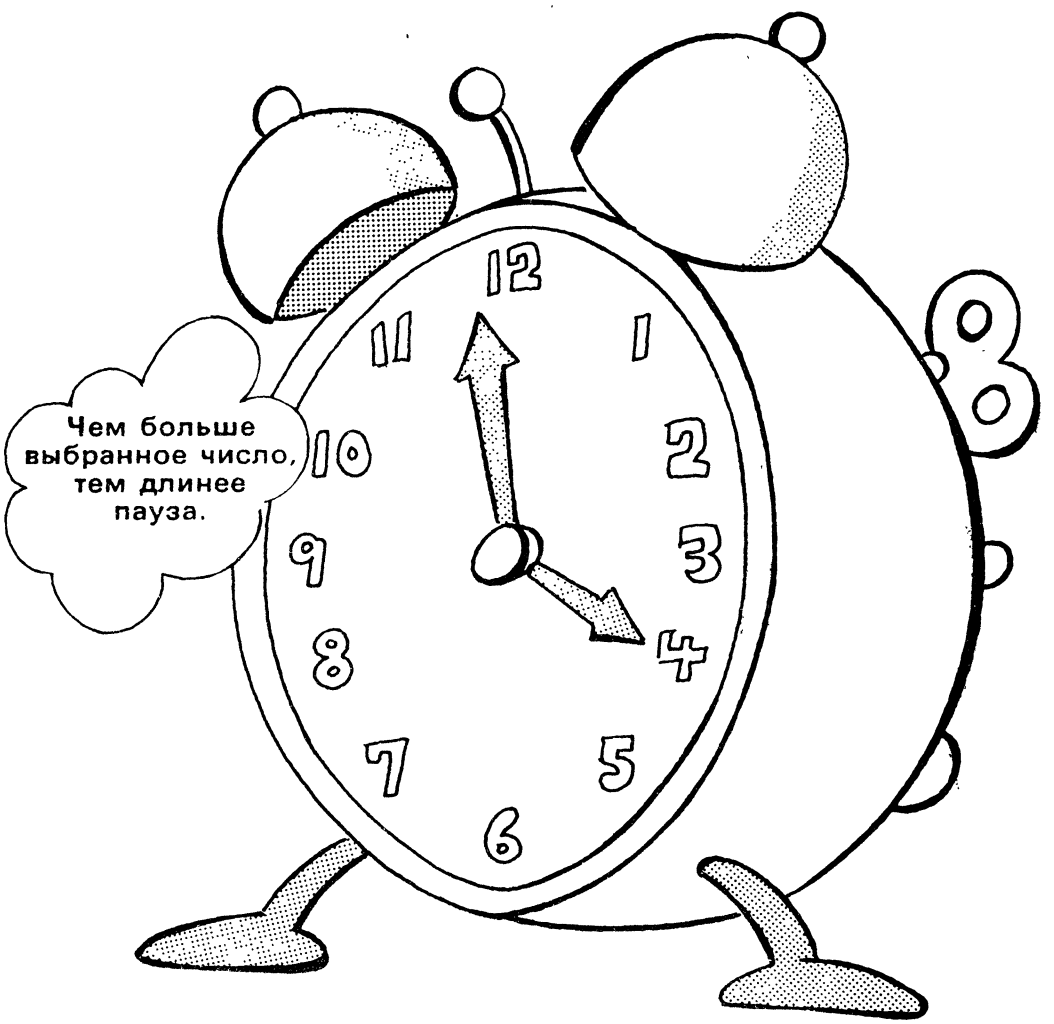
```
CLS:NEW
10 ? «ПРОВЕРИМ НАШУ ТАБЛИЦУ
     УМНОЖЕНИЯ»
20 ? «НАЧНЕМ С УМНОЖЕНИЯ НА 3»
30 FOR A = 1 TO 12
40 LET B = 3 * A
50 ?3;"□X□";A;"□=□";B
60 NEXT A
70 END
```

Если тебе кажется, что компьютер работает слишком быстро, можешь замедлить его работу с помощью цикла **FOR/NEXT**. Сравни эти три программы, и тебе все станет ясно.

```
CLS:NEW
10 ? «ПРИВЕТ»
20?«КОПУША!»
30 END
```

```
CLS:NEW
10 ? «ПРИВЕТ»
20 FOR J = 1 TO 100
30 NEXT J
40 ?«КОПУША!»
50 END
```

```
CLS:NEW
10 ? «ПРИВЕТ»
20 FOR J = 1 TO 1000
30 NEXT J
40 ? «КОПУША!»
50 END
```



. . . STEP . . .

Помнишь, как ты заставлял компьютер считать от 5 через 5, от 10 через 10 и т. д.? Можно получить тот же самый результат, используя цикл **FOR/NEXT**. Чтобы компьютер хорошо понимал, с каким шагом он должен действовать, надо использовать слово **STEP**. Смотри:

```
CLS:NEW
10 FOR A = 0 TO 100 STEP 5
20 PRINT A
30 NEXT A
40 END
```

Видишь как легко?

А можно заставить компьютер считать «назад». Как? Очень просто: надо только заданную величину шага сделать отрицательной, например равной −5. Попробуй – увидишь!

Давай проверим, как ты меня понял. Попроси компьютер сосчитать:

с шагом 2 до 100;

с шагом 10 от 100 до 0;

с шагом 3 до 100 (конечно, он остановится на каком-то числе, которое не точно равно 100).

Вложенные циклы

Ты можешь вкладывать циклы один в другой сколько угодно раз, лишь бы они не пересекались. Вот тебе один пример:

```
CLS:NEW
10 ?«МНЕ НРАВИТСЯ ХОДИТЬ В ШКОЛУ»
20 FOR B = 1 TO 10
30 ? «А ВЕДЬ ЭТО НЕПРАВДА!»
40 FOR A = 1 TO 5
50 ? «НЕТ, ПРАВДА!»
60 NEXT A
70 NEXT B
80 END
```

А теперь испытай такую программу:

```
CLS:NEW
10 ? «ВНИМАНИЕ,  ОТСЧИТЫВАЕМ
     ВРЕМЯ  ДО  ЗАПУСКА»
20 ?:?
30 FOR N = 10 TO 0 STEP -1
40 FOR K = 1 TO 300
50 NEXT K
60 ?N;"□ «СЕКУНД (S)»
70 NEXT N
80 ? «ЗАЖИГАНИЕ»
90 FOR J = 1 TO 250
100 NEXT J
110 FOR B = 1 TO 30
120 ?"        X        "
130 ?"      X   X      "
140 ?"   X    X    X   "
150 ?"   X    X    X   "
160 ?"   X    X    X   "
170 ?"   X    X    X   "
180 ?"   X    X    X   "
190 ?"   X    X    X   "
200 ?"   X    X    X   "
210 ?"X X          X X"
220 ?"X X          X X"
230 NEXT B
240 ?«УСПЕШНЫЙ ЗАПУСК!»
250 END
```

ЗВУКИ

Запиши две небольшие программы:

```
CLS:NEW
10 BEEP
20 GOTO 10
30 END
```

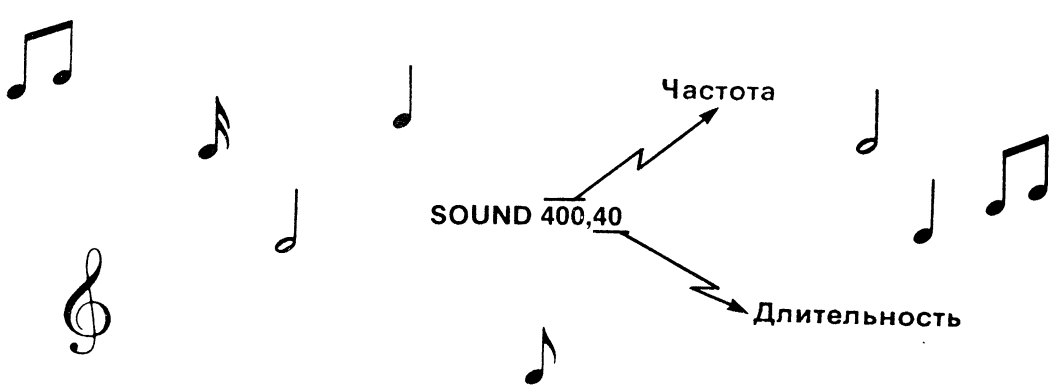
```
CLS:NEW
10 FOR A = 1 TO 500
20 BEEP
30 NEXT A
40 END
```

Но что это? Машина подает голос? Ты, наверное, раньше не знал, что она умеет петь? Перепиши эти две программы и внимательно прислушайся.

```
CLS:NEW
10 SOUND 400,20
20 SOUND 400,40
30 SOUND 400,60
40 END
```

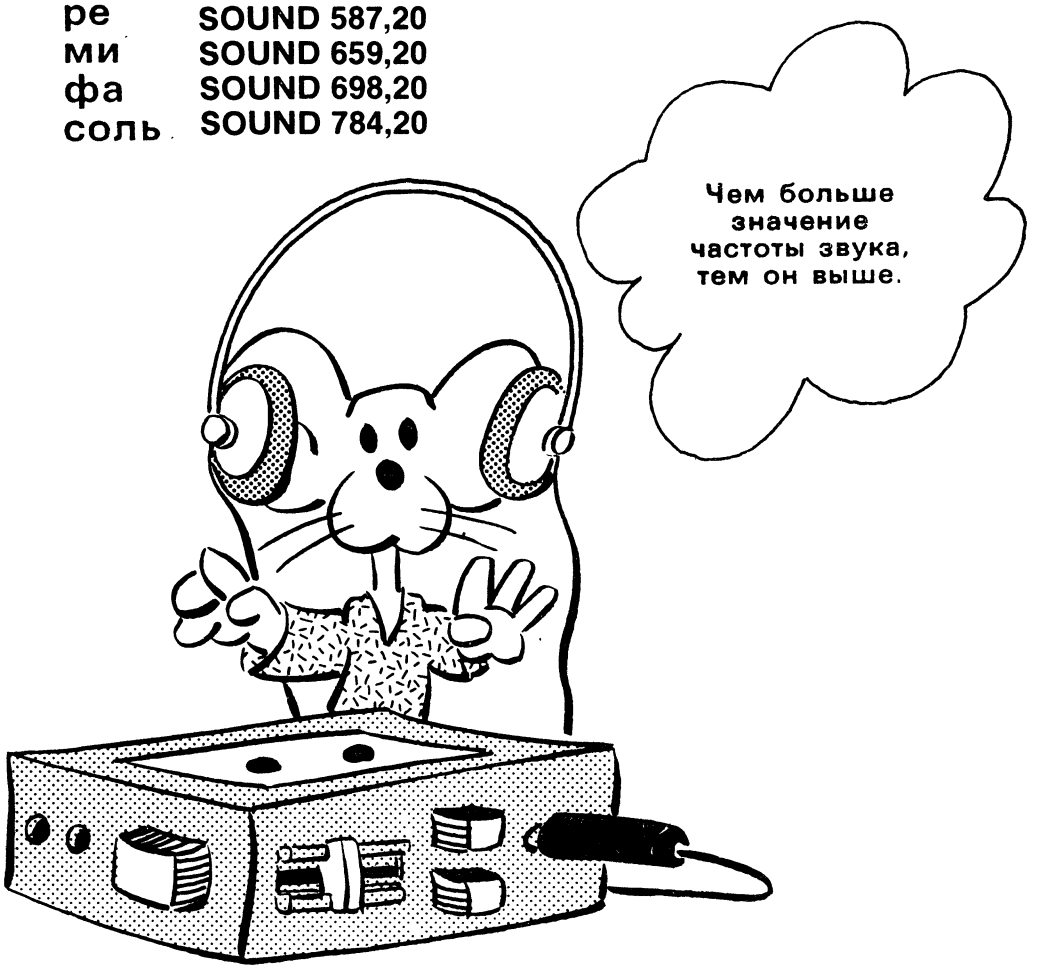
```
CLS:NEW
10 FOR A = 100 TO 2000 STEP 50
20 SOUND A,20
30 NEXT A
40 END
```

Команда **BEEP** заставляет компьютер издавать звуки, в то время как более сложная инструкция **SOUND** заставляет его петь. Два числа, которые следуют за инструкцией **SOUND**, очень важны: первое определяет частоту звука, второе – его длительность.

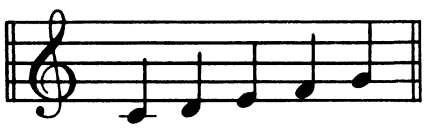


Можешь развлечься, набирая любые ноты, какие захочешь. Вот, например, пять первых нот, входящих в гамму «до»:

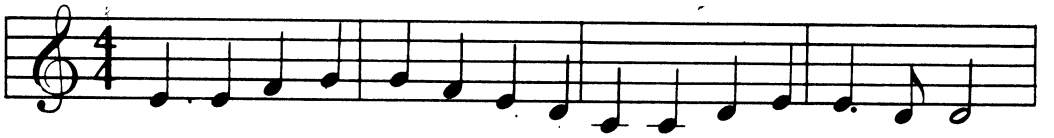
до	SOUND 523,20
ре	SOUND 587,20
ми	SOUND 659,20
фа	SOUND 698,20
соль	SOUND 784,20



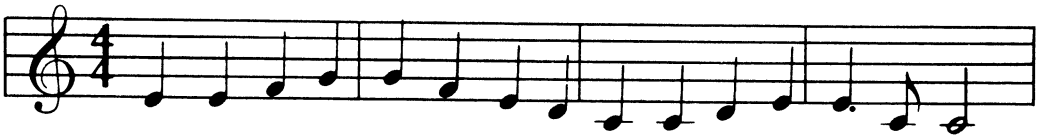
А теперь попробуй сам сочинить какую-нибудь мелодию.



«Ода к Радости»
(Бетховен, Девятая симфония)

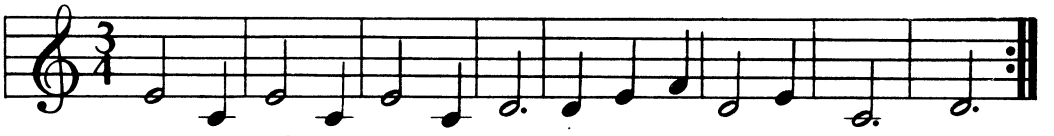


mi mi fa sol sol fa mi ré do do ré mi mi ré ré

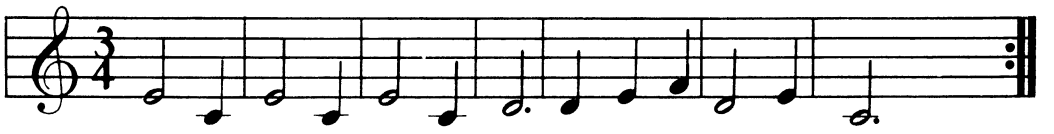


mi mi fa sol sol fa mi ré do do ré mi ré do do

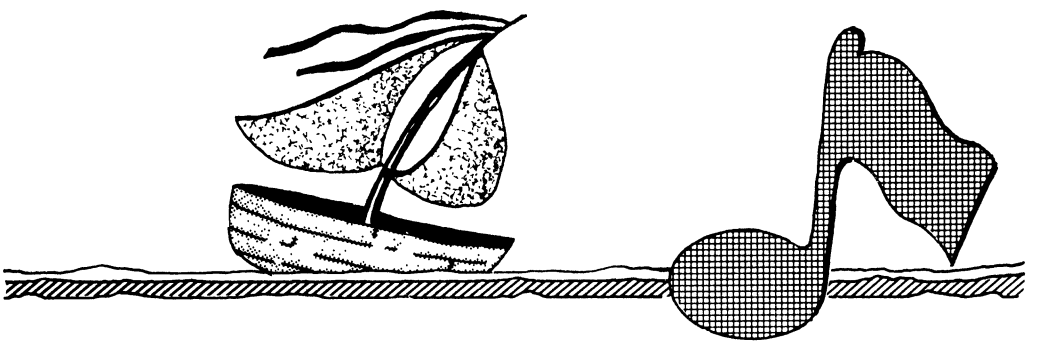
«Веселый ручеек»
(Ги Беар)



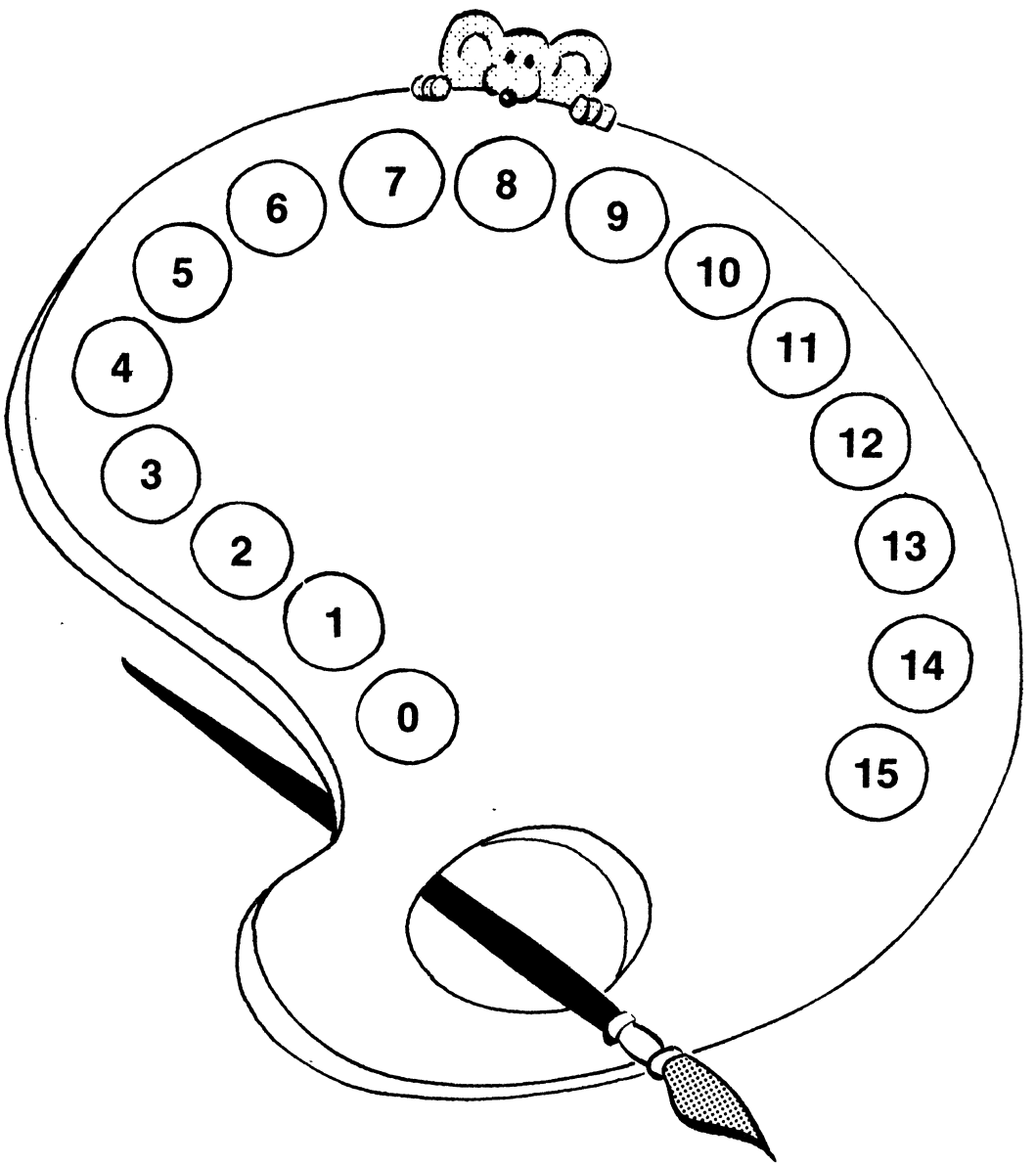
mi do mi do mi do ré ré mi fa ré mi do ré



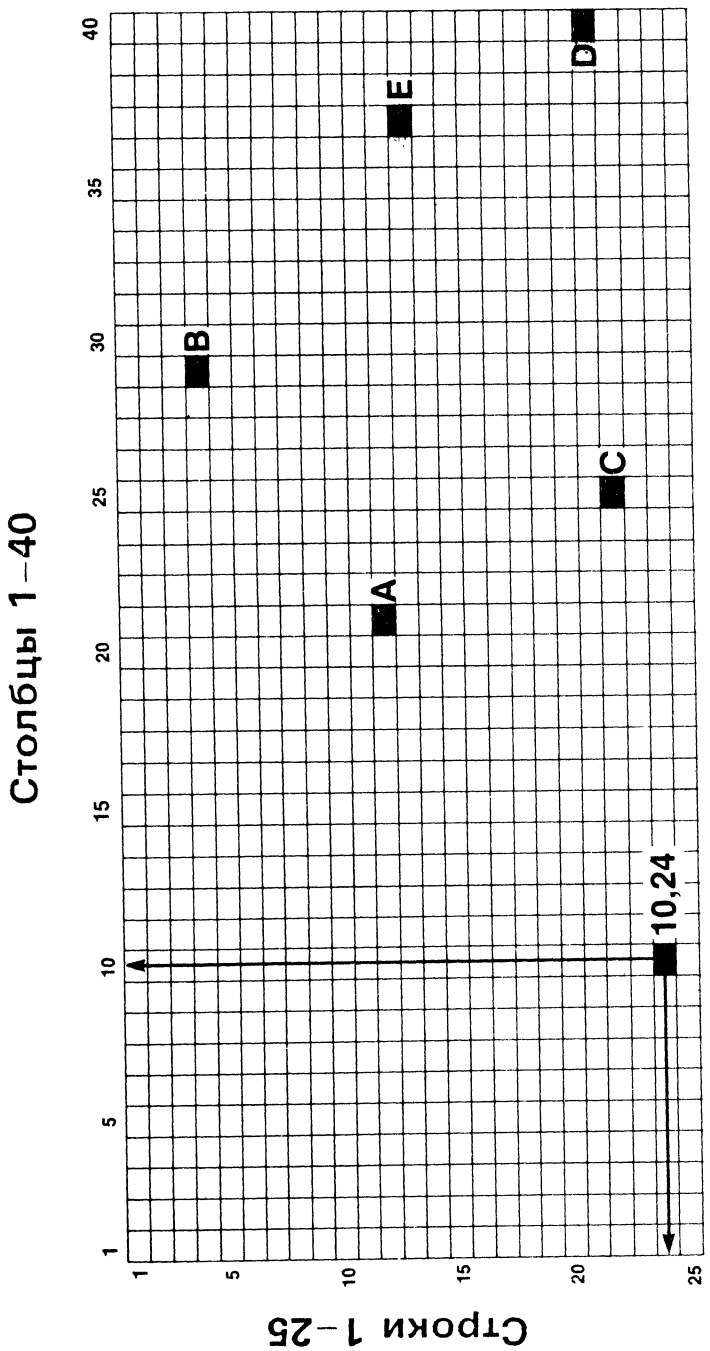
mi do mi do mi do ré ré mi fa ré mi do



КАРТИННАЯ ГАЛЕРЕЯ



КООРДИНАТЫ



Посмотри на координатную сетку, изображенную на предыдущей странице. Видишь, клеточка с координатами 10, 24 расположена в 10-м столбце на 24-й строке.

На сетке показаны еще пять клеточек, обозначенных буквами A, B, C, D, E. Запиши их координаты в таблицу.

	Столбец	Строка
A		
B		
C		
D		
E		

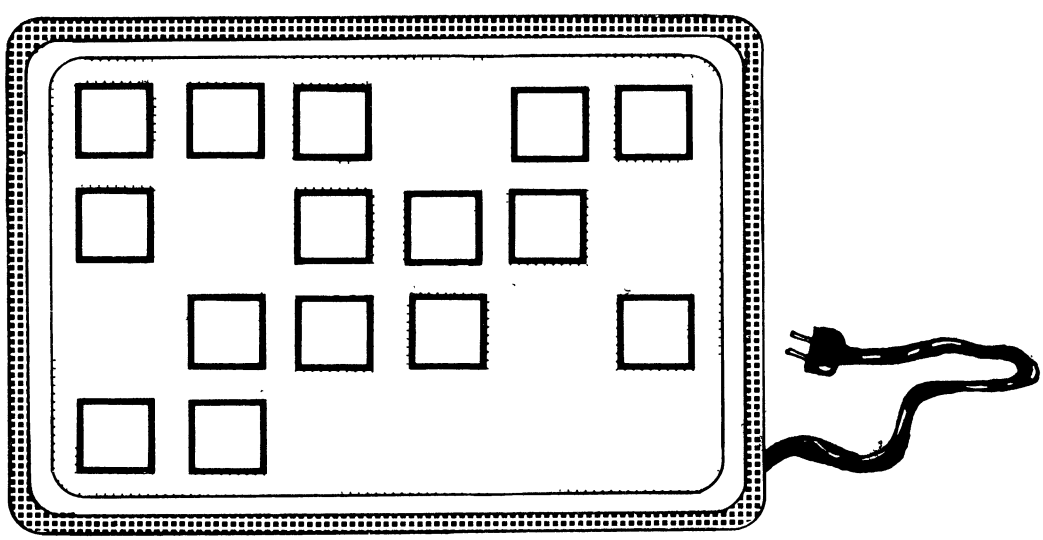
С помощью системы координат можно расположить букву, цифру или любой символ в какой угодно точке экрана. Для этого достаточно только правильно указать компьютеру адрес.

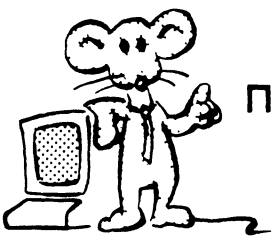
Когда ты работаешь с текстом (*текстовый режим*), твой экран состоит из 40 (**WIDTH 40**) или 80 (**WIDTH 80**) столбцов, по 25 строк в каждом. Адреса, которые ты сообщаешь компьютеру, должны соответствовать *экранной сетке*, обозначаемой **SCREEN 0: WIDTH 40** (или **80**).

Сообщить компьютеру адрес ты можешь с помощью инструкции **LOCATE**, за которой идут числа, разделяемые запятыми. Будь внимателен! Когда компьютер работает в текстовом режиме, инструкция **LOCATE** требует, чтобы вначале был указан номер строки, а затем столбца. Например, если ты укажешь

```
CLS:NEW
10 SCREEN 0:WIDTH 40:KEY OFF
20 LOCATE 13,20
30 ?"*"
40 END
```

то получишь звездочку (*) в центре экрана, т. е. на пересечении строки 13 и столбца 20.





Поиграем с координатами

1. Посмотри, какой получится забавный рисунок, если ты поместишь звездочки в точки экрана, обозначенные следующими координатами:

6,16	18,18	15,19	6,15	12,12	16,23	19,19
6,22	16,16	6,19	6,21	12,26	17,17	14,17
7,14	16,22	5,19	6,23	14,13	17,21	14,21
7,24	10,16	4,20	8,13	14,25	3,17	15,20
15,14	10,22	4,18	8,25	16,15	2,16	15,18
15,24	11,16	7,19	10,12	1,23	2,22	12,19
18,20	11,22	6,17	10,26	1,15	3,21	

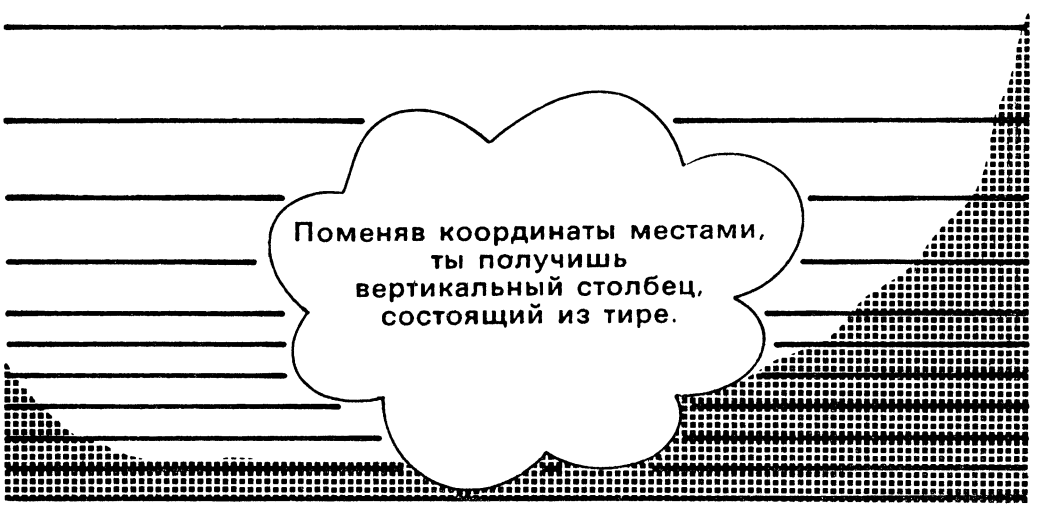
2. Нарисуй на клетчатой бумаге какой-нибудь рисунок, а затем воспроизведи его на экране. Не забудь в начале программы написать **SCREEN** 0: **WIDTH 40**(или **80**): **KEY OFF**.

СТРОКИ

Ты уже пытался рисовать горизонтальную строчку на экране? Если да, то должен был заметить, что это достаточно длительное занятие. Перепиши следующую программу и увидишь, что с помощью цикла **FOR/NEXT** все можно сделать намного проще.

```
CLS:NEW
10 SCREEN 0:WIDTH 40:KEY OFF
20 FOR L = 10 TO 20
30 LOCATE 18,L
40 ?"—"
50 NEXT L
60 END
```

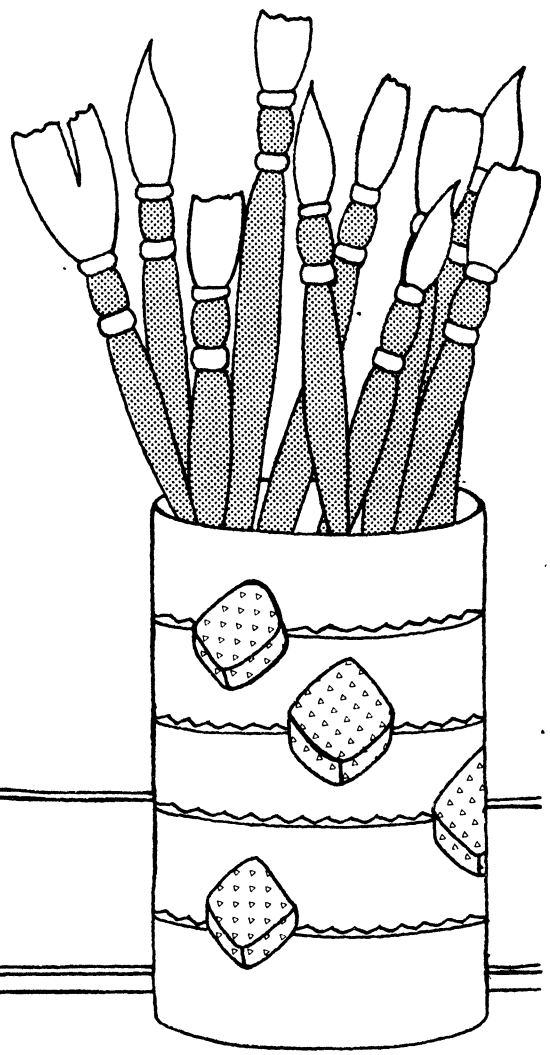
Поскольку в цикле **FOR/NEXT** переменная L принимает значение от 10 до 20, а сама величина L – это вторая координата в инструкции **LOCATE**, то компьютер изобразит тире (–) на позициях 18, 10 – 18, 11 – 18, 12 – 18, 13 – 18, 14 – 18, 15 – 18, 16 – 18, 17 – 18, 18 – 18, 19 – 18, 20.



ЦВЕТА

Информатика — очень красочная страна. Чтобы убедиться в этом, перепиши аккуратно следующую программу. Не забывай нажимать на клавишу **ENTER** (↵) в конце каждой строки. Но до этого исправь допущенные ошибки.

```
CLS:NEW
10 SCREEN 0:WIDTH 40:KEY OFF
20 COLOR 1
30 LOCATE 5,20
40 PRINT "*"
50 COLOR  2
60 LOCATE 5,36
70 PRINT "*"
80 COLOR 3
90 LOCATE 21,20
100 PRINT "*"
110 COLOR 4
120 LOCATE 21,36
130 PRINT "*"
140 COLOR 5
150 LOCATE 13,28
160 PRINT "*"
170 END
RUN
```



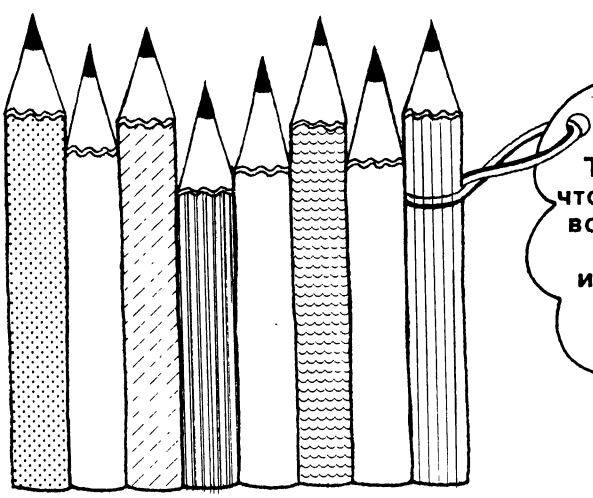
Заметь, что английское слово COLOR, означающее ЦВЕТ, похоже на русское слово КОЛЕР.

Те пять цветов, которые ты только что изобразил на экране, далеко не все, что может компьютер. Честное программышское, цветов еще очень много!

Чтобы компьютеру было легко находить нужный цвет, каждому из них нужно присвоить номер. Всего имеется **16 цветов**, пронумерованных от 0 до 15. Вот они:

ЧЕРНЫЙ = 0	ТЕМНО-СЕРЫЙ = 8
СИНИЙ = 1	СВЕТЛО-СИНИЙ = 9
ЗЕЛЕНЫЙ = 2	СВЕТЛО-ЗЕЛЕНЫЙ = 10
ГОЛУБОЙ = 3	СВЕТЛО-ГОЛУБОЙ = 11
КРАСНЫЙ = 4	РОЗОВЫЙ = 12
ПУРПУРНЫЙ = 5	СВЕТЛО-ПУРПУР-НЫЙ = 13
ЗОЛОТИСТЫЙ (КОРИЧНЕВЫЙ) = 6	ЖЕЛТЫЙ = 14
БЕЛЫЙ (СЕРЫЙ) = 7	ЯРКО-БЕЛЫЙ = 15

Если ты хочешь увидеть на экране какие-нибудь из этих цветов, проделай снова упражнение, заданное на стр. 89, изменив номера цветов.

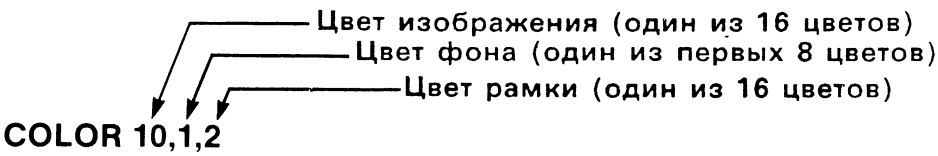


Теперь, когда ты знаешь, что означают номера цветов, возьми цветные карандаши и раскрась палитру, изображенную на рисунке на стр. 83 .

Кроме цвета того предмета, который хочешь отобразить, ты можешь еще выбрать цвет фона и цвет рамки. Написав, например,

```
COLOR 14,1,12
```

ты получишь желтое изображение на синем фоне в розовой рамке. Порядок следования номеров в записи очень важен: первое число определяет цвет изображения, второе – цвет фона, а третье – цвет обрамления.



Напиши следующую программу:

```
CLS:NEW
10 SCREEN 0:WIDTH 40:KEY OFF
20 COLOR 15,5,5:CLS
30 FOR R = 1 TO 10
40 ?" «Я ВИЖУ ЖИЗНЬ В
       РОЗОВОМ СВЕТЕ»
50 NEXT R
60 END
RUN
```

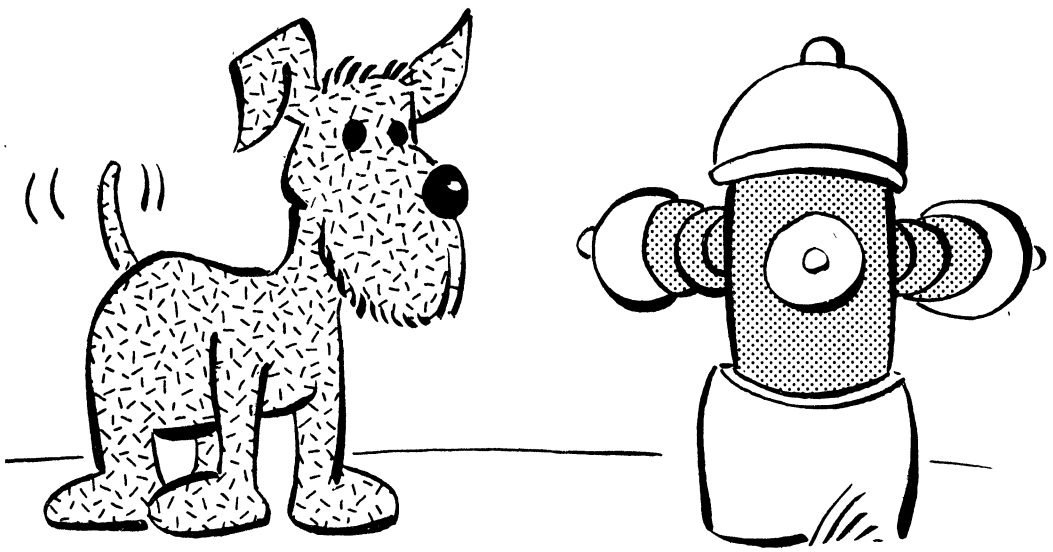
Для получения нужного цвета надо хорошо очистить экран (CLS) перед тем как дать инструкцию COLOR.

и посмотри, что получится, если заменить строки 20 и 40 на следующие:

```
20 COLOR 15,1,1:CLS
40 ? «ПРОГРАММЫШКА СМЕРТЕЛЬНО
      БОИТСЯ КОШЕК»
```

Можешь сам поразвлекаться с цветами.

Теперь ты уже можешь создавать движущиеся рисунки. Но сначала посмотри, что я для тебя сочинила.



Прогулка Шарика

```
CLS:NEW
10 SCREEN 0:WIDTH 40:KEY OFF
20 COLOR 14,1,1:CLS
30 LOCATE 6,38:?"X"
40 LOCATE 6,39:?"X"
50 LOCATE 6,40:?"X"
60 LOCATE 5,39:?"X"
70 LOCATE 7,39:?"X"
80 LOCATE 8,39:?"X"
90 LOCATE 9,39:?"X"
```

Водоразборная колонка

```
100 A = 1
110 COLOR 15
120 A = A + 1
130 C = A - 1
140 LOCATE 10,A:?"*"
```

Шарик

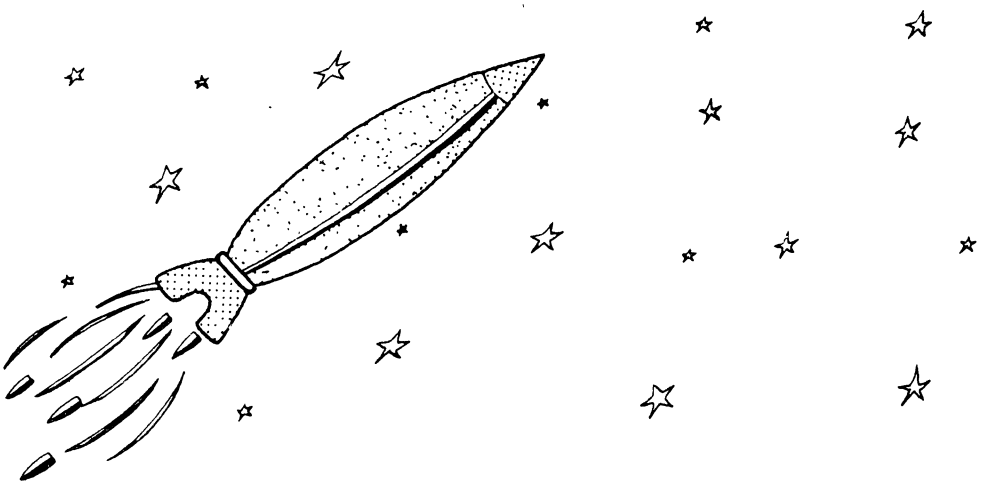
```
150 FOR B = 1 TO 100:NEXT B
```

Цикл для создания паузы

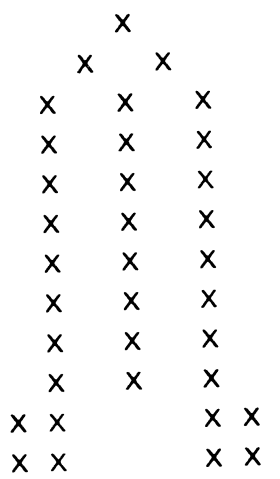
```
160 COLOR 1
170 LOCATE 10,C:?"  "
```

Изображение черной точки, чтобы создать впечатление, будто Шарик пропал

```
180 IF A = 37 THEN 200          Движение Шарика к колонке
190 GOTO 110
200 COLOR 15
210 A = A - 1
220 C = A + 1
230 LOCATE 10,A:?"*"
240 FOR B = 1 TO 100:NEXT B     Возвращение Шарика домой
250 COLOR 1
260 LOCATE 10,C:?"  "
270 IF A = 1 THEN END
280 GOTO 200
```



А сейчас действуй сам. Попытайся, например, запустить ракету.



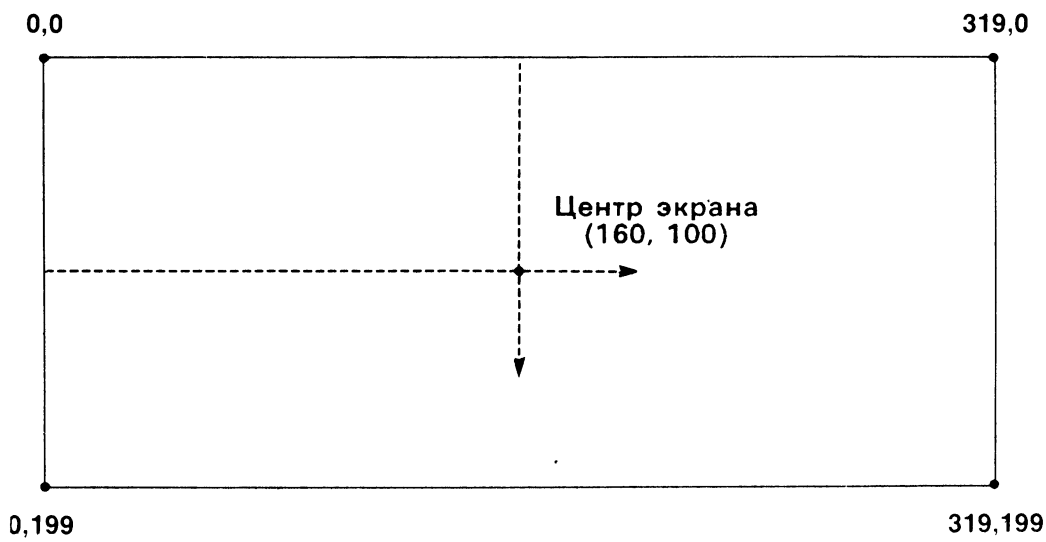
ГРАФИЧЕСКИЙ РЕЖИМ

Ты, наверное, помнишь, как я познакомила тебя с группой художников (мы с ними встретились в конце первой части на страницах 35–38). Да, да, это именно они – **U, R, D** и **L** – дали тебе первые уроки компьютерного рисования. Что бы ты сказал, если бы я предложила тебе снова встретиться с ними?

Приготовься к встрече. Для этого напиши на экране:

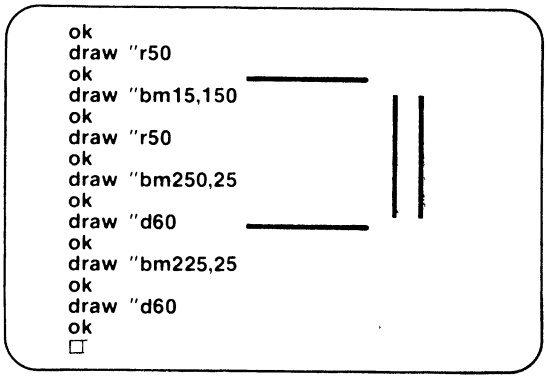
```
SCREEN 1:KEY OFF:CLS
```

Ты теперь уже знаешь, какую роль выполняют команды **KEY OFF** и **CLS**. А вот для чего нужна инструкция **SCREEN 1**? Да точно так же, как **SCREEN 0**, эта инструкция служит для определения режима работы и размера сетки экрана. На этот раз мы имеем графический режим, а экран состоит из 320 столбцов (с номерами 0 – 319) и 200 строк (с номерами 0 – 199).



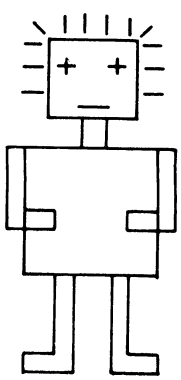
Перед тем как начать рисовать, аккуратно перепиши следующую программу:

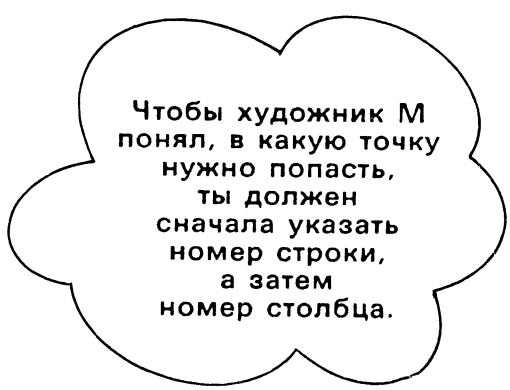
```
DRAW "R50
DRAW "BM15,150
DRAW "R50
DRAW "BM250,25
DRAW "D60
DRAW "BM225,25
DRAW "D60
```



Ты уже раньше познакомился со всеми художниками, кроме **M**. Роль этого нового знакомого заключается в том, чтобы просить компьютер поместить карандаш или перо в совершенно определенную точку, координаты которой написаны после буквы **M**. В данном случае перемещение будет невидимым, поскольку перед **M** стоит **B**.

Благодаря этому новому другу ты можешь теперь начинать свой рисунок там, где захочешь. Для тренировки можешь нарисовать вот такого маленького робота.

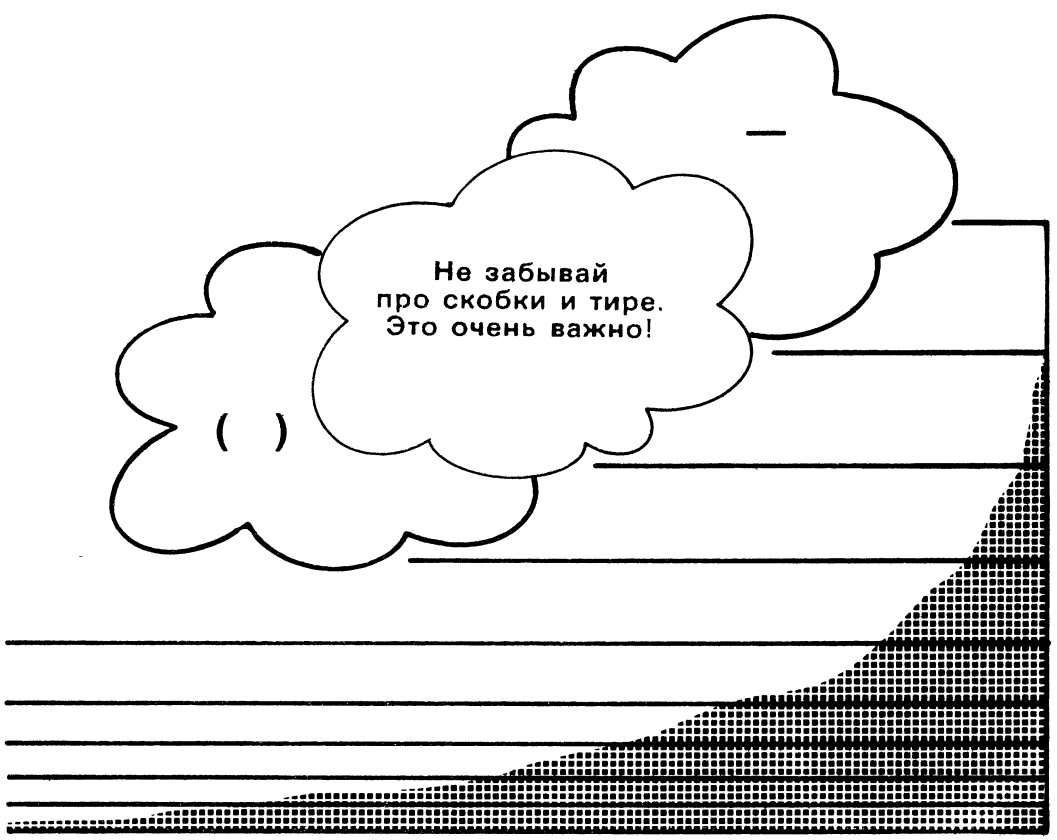




Существует и другой способ изображения линий. На этот раз ты должен использовать инструкцию **LINE** для того, чтобы сказать компьютеру, в какой точке отрезок начнется и в какой закончится.

Очисти экран и набери следующую программу:

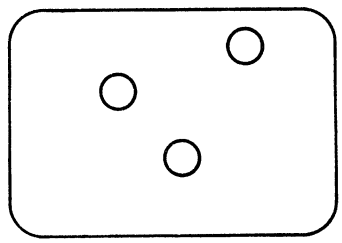
```
SCREEN 1:KEY OFF
LINE (200,100) — (300,100)
LINE (300,100) — (300,150)
LINE (300,150) — (200,150)
LINE (200,150) — (200,100)
```



ОКРУЖНОСТИ

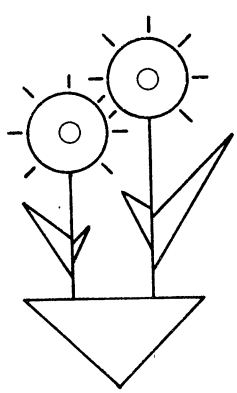
В графическом режиме можно рисовать на экране окружности.

```
CLS:NEW
10 SCREEN 1:KEY OFF
20 CIRCLE (200,65),25
30 CIRCLE (100,90),25
40 CIRCLE (150,130),25
50 END
RUN
```



Как видишь, достаточно задать компьютеру после инструкции **CIRCLE** координаты центра в скобках и длину радиуса окружности.

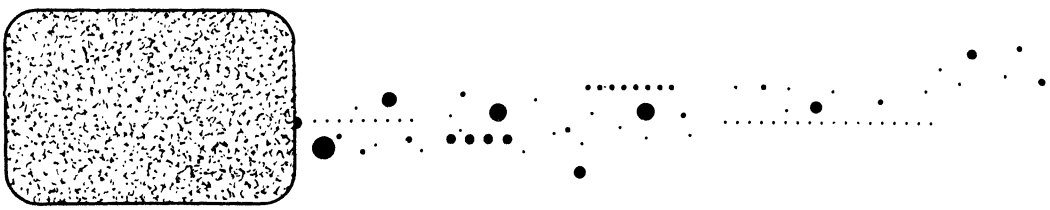
Сначала нарисуй на экране несколько окружностей, а затем, комбинируя прямолинейные отрезки и окружности, нарисуй вот такие цветы в горшочке:



А теперь давай посмотрим, как же можно заставить точки появляться и исчезать на экране.

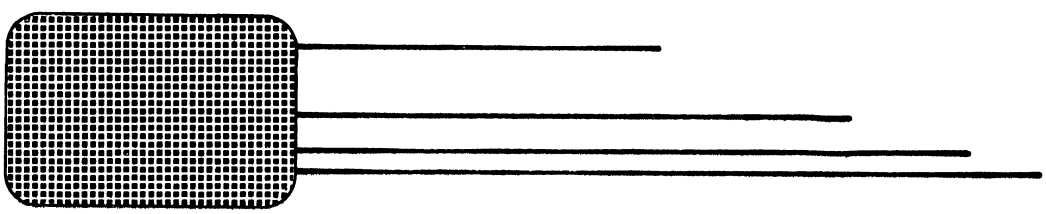
```
CLS:NEW
10 SCREEN 1:KEY OFF
20 PSET (160,100)
30 FOR A = 1 TO 500:NEXT A
40 PRESET (160,100)
50 FOR A = 1 TO 500:NEXT A
60 END
RUN
```

Смотри, как это получается: сначала инструкция **PSET** требует, чтобы компьютер изобразил точку на экране в позиции с координатами (160, 100). Затем инструкция **PRESET** требует стереть ее. (Как ты знаешь, цикл **FOR/NEXT** служит только для создания паузы.)



А теперь постарайся стереть эту линию с помощью инструкции **PRESET**.

```
CLS:NEW
10 SCREEN 1:KEY OFF
20 FOR X = 160 TO 200
30 PSET (X,100)
40 NEXT X
50 END
RUN
```



Инструкции **PSET** и **PRESET** могут служить также для изображения линий на экране.

Все эти линии, точки и кружочки были бы значительно более интересны в раскрашенном виде, не правда ли? Ну так что же, почему бы не попытаться их раскрасить?

В графическом режиме можно раскрасить экран, выбирая один из шестнадцати цветов, с которыми мы с тобой уже знакомились, рассматривая текстовый режим (стр. 92). Однако сейчас в твоем распоряжении только шесть цветов, разделенных на две палитры:

ЦВЕТ	ПАЛИТРА 0	ПАЛИТРА 1
1	зеленый	синий
2	красный	пупурный
3	золотистый (коричневый)	белый

Рассмотрим пример:

```
CLS:NEW
10 SCREEN 1:KEY OFF
20 COLOR 9,0:CLS
```

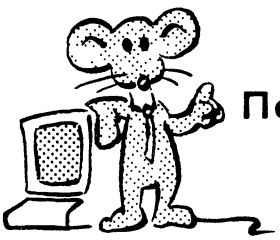
Цвет фона — Палитра

```
30 LINE (160,100) — (160,200),3
40 END
```

Цвет изображения

Текст на строке 20 определяет цвет экрана (9 = светло - синий) и говорит компьютеру о том, что нужно выбрать цвет изображения на *палитре 0.*

Текст на строке 30 требует от компьютера провести линию и окрасить ее в золотисто-коричневый цвет (третий цвет на *палитре 0*).

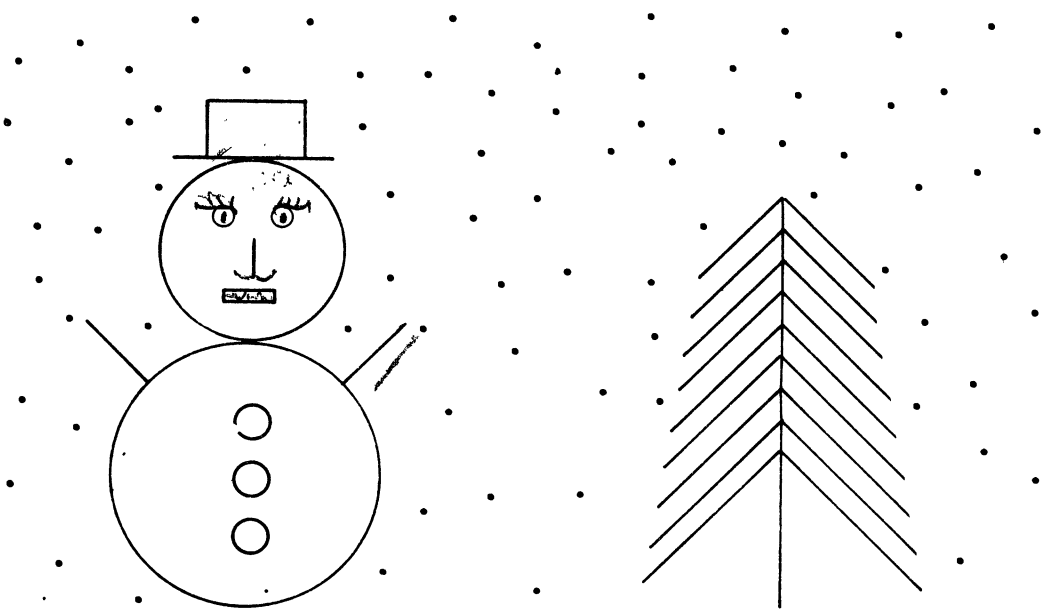


Поразвлекайся

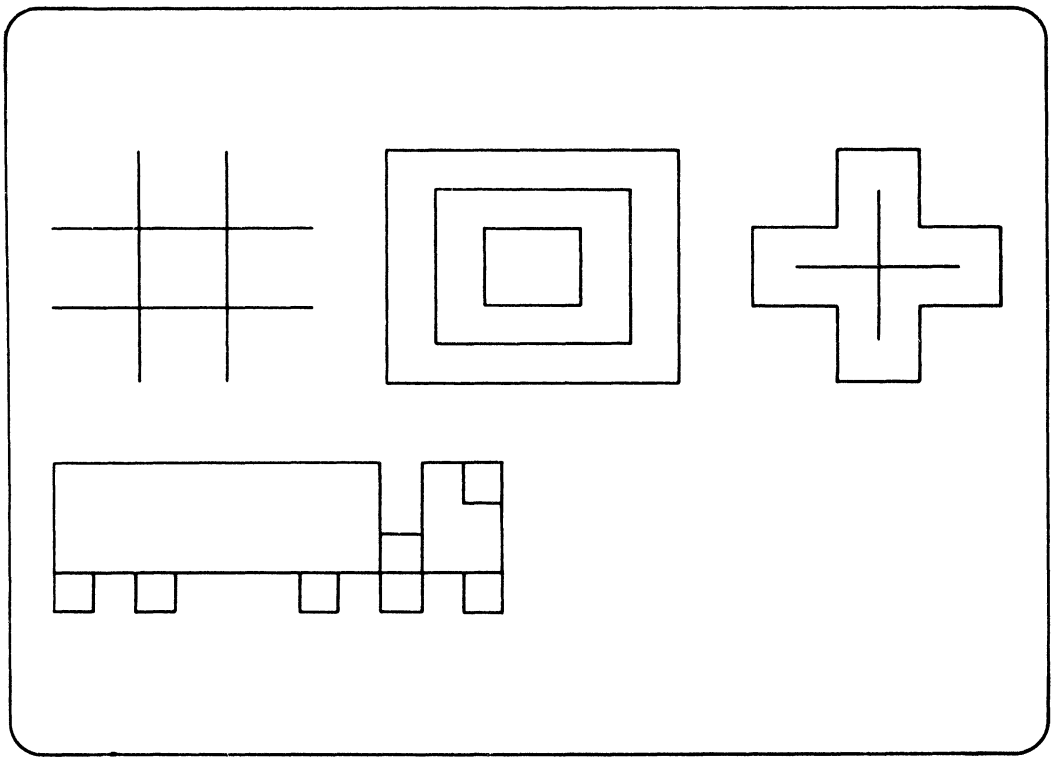
1. Нарисуй в каждом углу экрана маленькую коробочку.

2. Добавь по кружочку к каждой коробочке, которые ты только что нарисовал.

3. На голубом фоне нарисуй снеговика и елку, а потом добавь немного снега.



4. Изобрази на экране следующие рисунки:

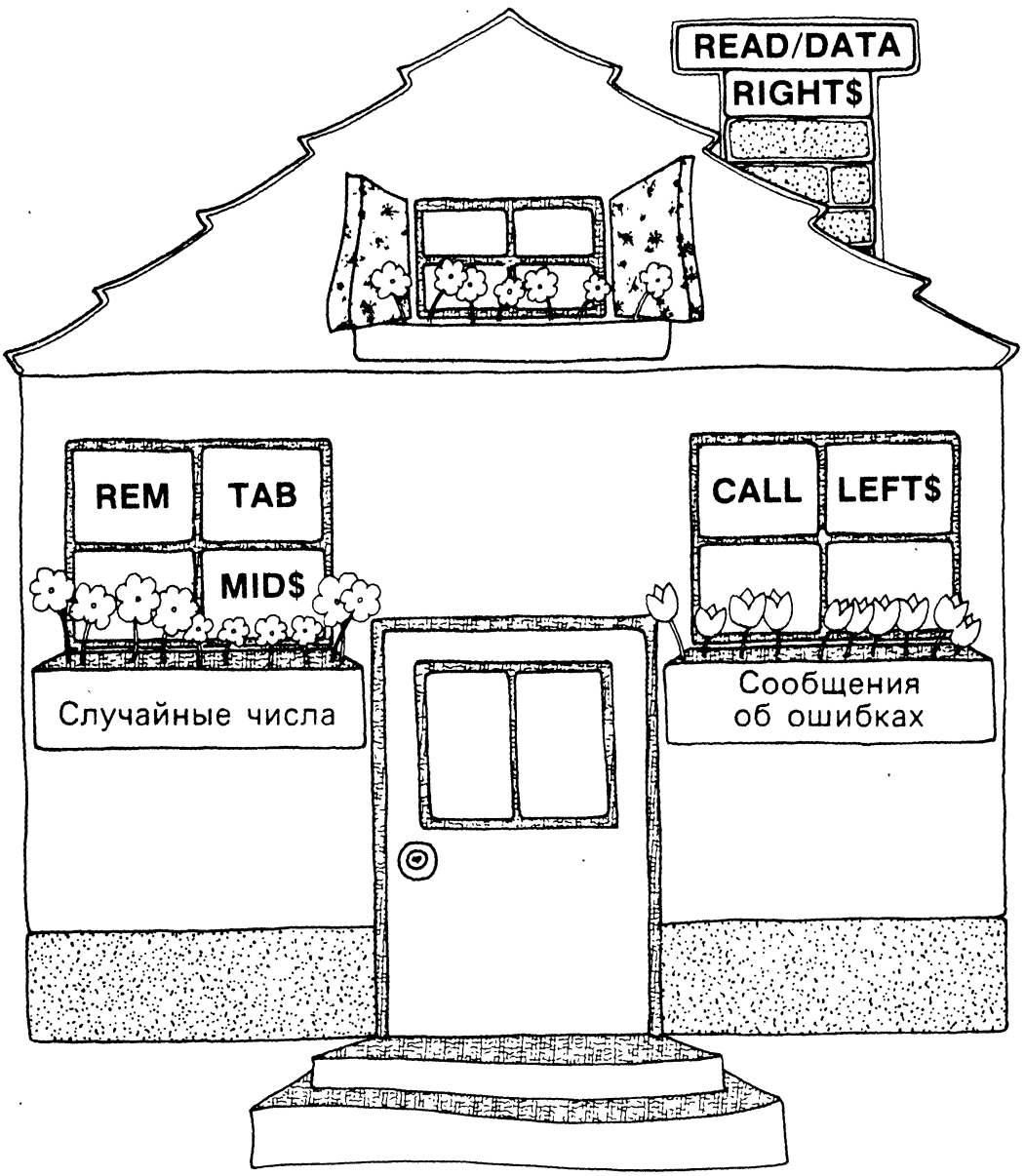


5. Напиши на экране свои инициалы.

6. Придумай сам какой-нибудь забавный рисунок и изобрази его.

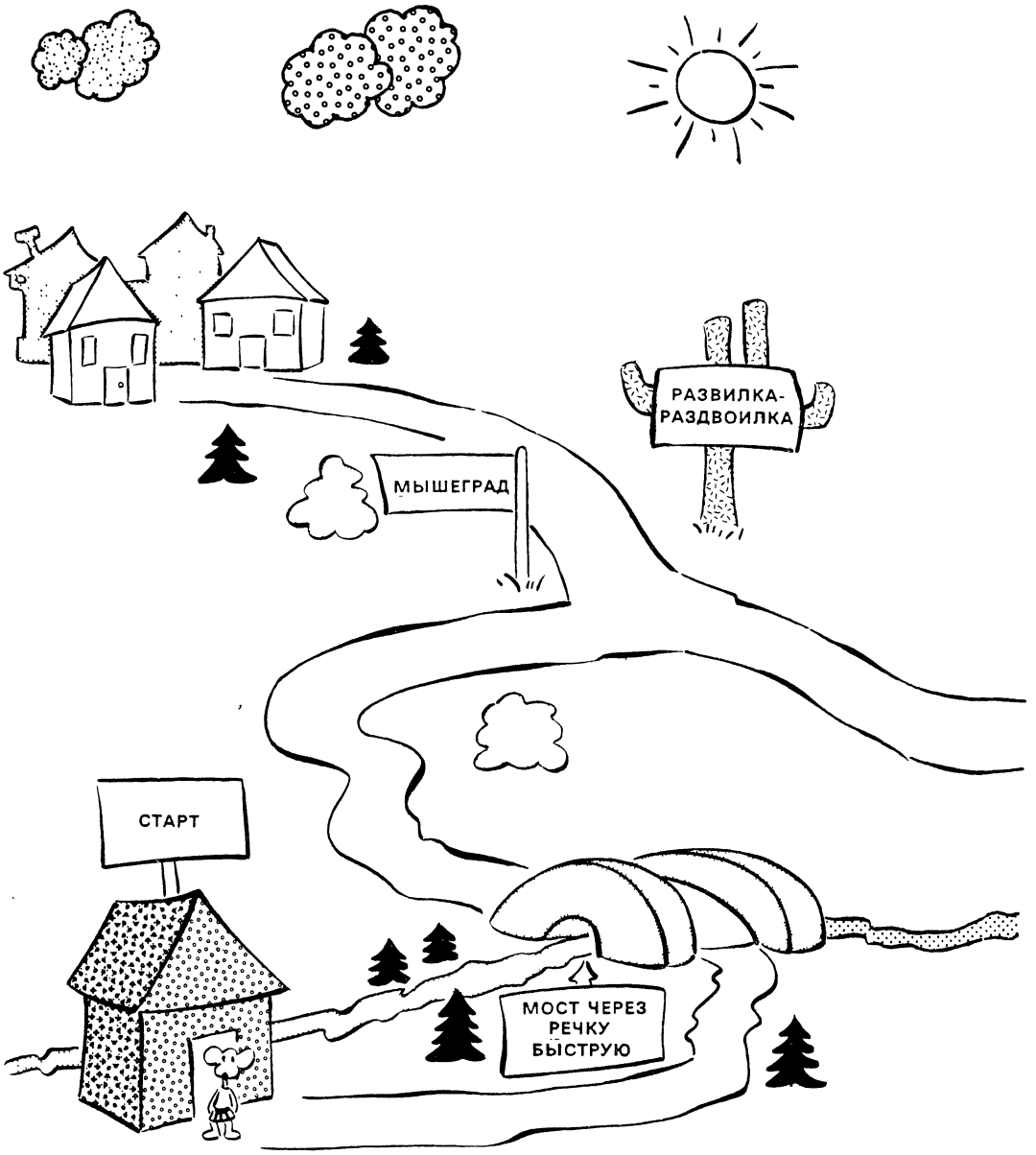
ТРЕТЬЯ ЧАСТЬ

ПОСЛЕДНИЕ ШТРИХИ



Ну что, нравится тебе Информатика? В самом деле? Наверное, это потому, что ты хорошо делал свое дело.

Если ты останешься с нами до конца, то узнаешь еще кое-что полезное. Кто знает, может быть, с помощью новых знаний ты сумеешь сочинить свои собственные видеоигры.


РАЗВИЛКА-
РАЗДВОИЛКА
МЫШЕГРАД
СТАРТ
МОСТ ЧЕРЕЗ
РЕЧКУ
БЫСТРУЮ

Программа для компьютера должна быть такой же точной, как дорожная карта: если допустить ошибку хотя бы на одном этапе, путешественник рискует заблудиться. Например, если я хочу добраться до домика Мышки Хохотушки, а ты мне дашь только такие указания, какие написаны ниже, то я, вероятно, никогда к ней не попаду.

ВЫЙТИ ИЗ ДОМА

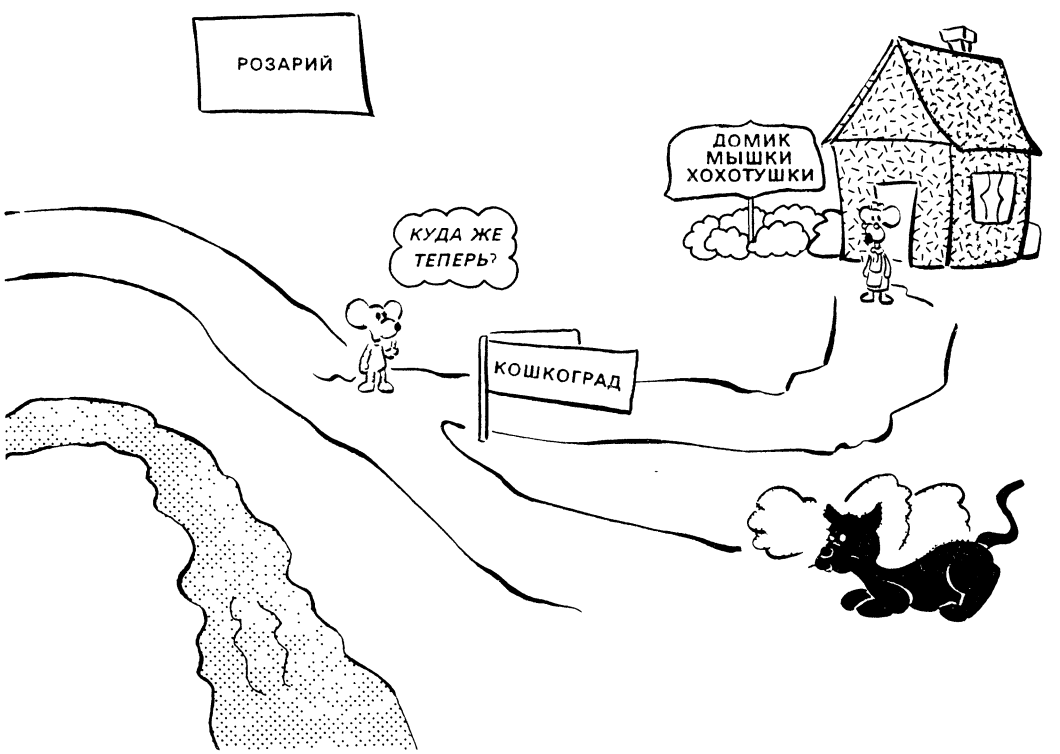
ПОВЕРНУТЬ НАПРАВО

ПЕРЕЙТИ РЕЧКУ БЫСТРУЮ ПО МОСТИКУ

НА РАЗВИЛКЕ-РАЗДВОИЛКЕ ПОВЕРНУТЬ НАПРАВО

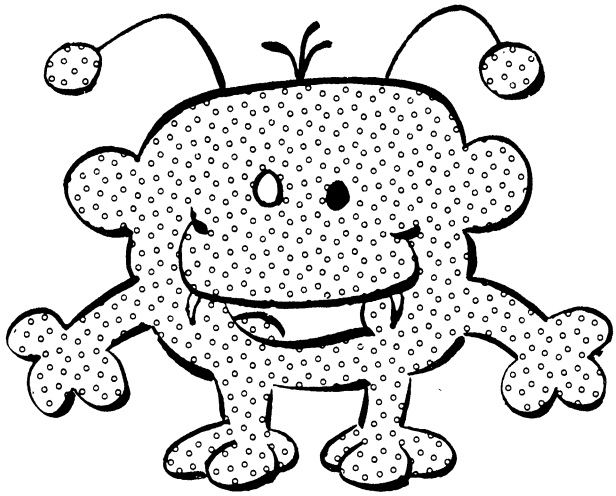
ПОЙТИ МИМО РОЗАРИЯ

ПРОДОЛЖАТЬ ИДТИ ПРЯМО ДО КОНЦА



ПРОГРАММНЫЕ «БЯКИ»

РАЗЫСКИВАЕТСЯ



СООБЩЕНИЕ ОБ ОШИБКЕ
по прозвищу «БЯКА»

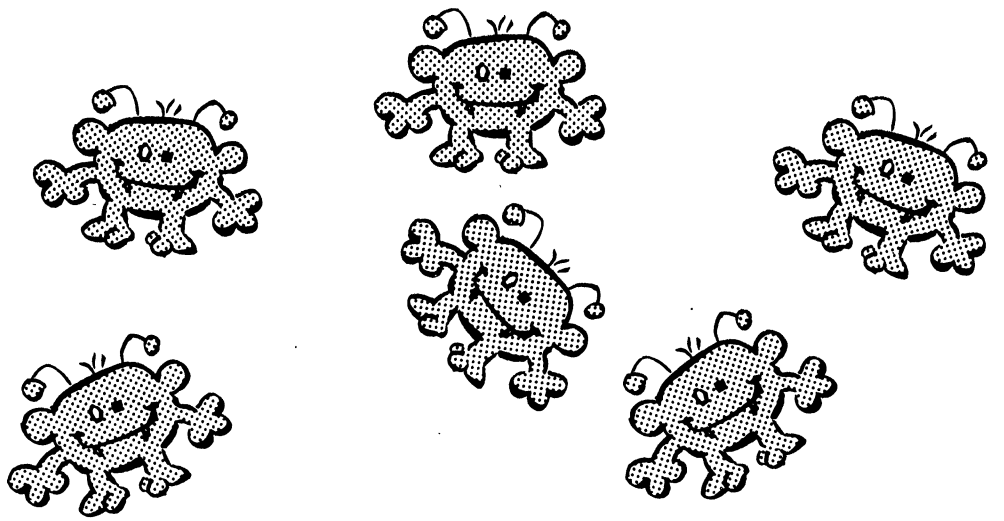
Характерные признаки

?SYNTAX ERROR
?OUT OF DATA ERROR
?NEXT WITHOUT FOR ERROR
?UNDEF'D STATEMENT ERROR

Каждый раз, когда составляешь программу, тебя подстерегают всевозможные ошибки. Чтобы их не допускать, надо быть очень внимательным. Если все же допущена ошибка, то это значит, что в программу забрался «Бяка», и тебе следует организовать его поиск, чтобы ошибку исправить. К счастью, «Бяка», часто оставляет следы.

?SYNTAX ERROR

?SYNTAX ERROR – это след, который наиболее часто остается после «Бяки». Такой след называется СООБЩЕНИЕ ОБ ОШИБКАХ. Если эти слова появляются на экране, то это означает, что ты допустил одну или несколько ошибок, например такого типа:



1. Орфографические ошибки.
   Ex.: **10 IMPUT ⇒ 10 INPUT**
2. Неправильное нажатие клавиши.
   Ex.: **10 A=1 TO 1O ⇒ 10 A=1 TO 10**
3. Ошибка в имени переменной.
   Ex.: **10 A= «ЗДРАВСТВУЙ» ⇒ 10 A$= «ЗДРАВСТВУЙ»**
4. Пропущенные кавычки.
   Ex.: **10 LET B$= ПРИВЕТ ⇒ 10 B$= «__ПРИВЕТ__»**
5. Использование несуществующей инструкции.
   Ex.: **10 FOR A=1+35 ⇒ 10 FOR A=1 TO 35**

Как только ты исправишь ошибку, «Бяка» исчезнет. В приложении Б ты найдешь и другие примеры сообщений об ошибках.

REM

Ты можешь сделать программу более наглядной и удобной для использования, снабдив ее комментариями. Действительно, хорошая программа должна:

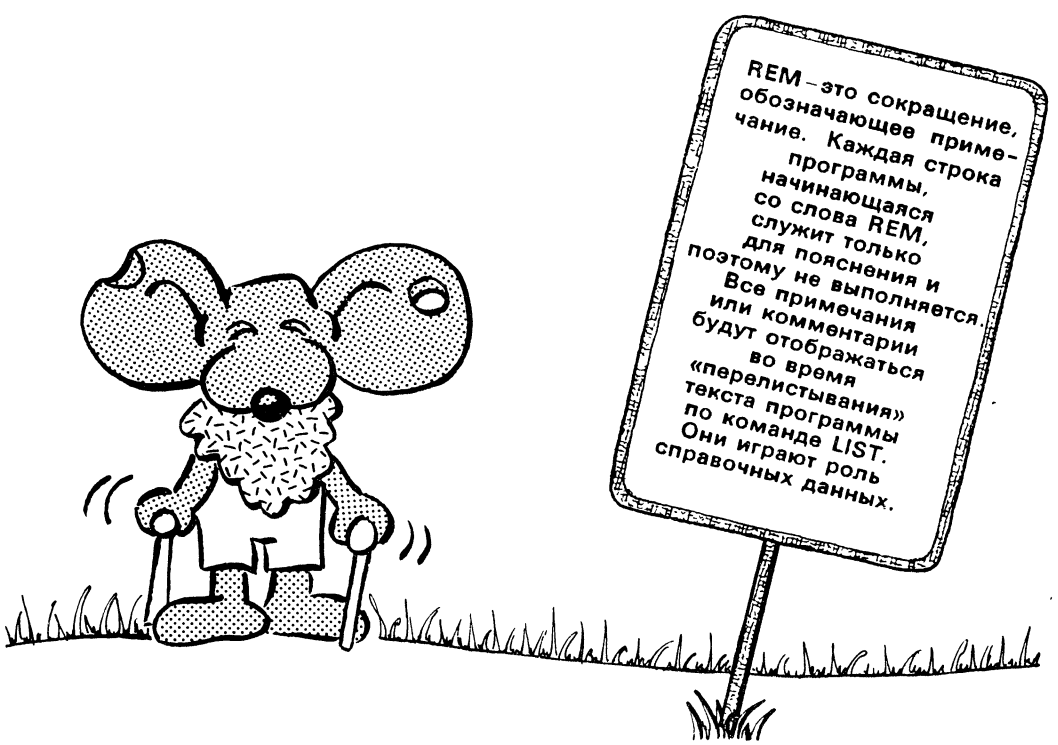
содержать перечень этапов выполнения задачи;

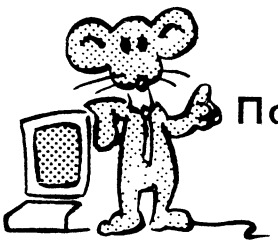
перечислять все этапы в нужной последовательности;

обозначать каждый этап с помощью инструкции **REM**.

Вот пример:

```
SCREEN 0:KEY OFF:NEW
10 REM ВЫЧИСЛЕНИЕ ВОЗРАСТА В ДНЯХ
20 ?«СКОЛЬКО ТЕБЕ ЛЕТ?»
30 INPUT A
40 REM ВЫПОЛНЕНИЕ ПРОГРАММЫ
50 D=365.25*A
60 ? «ЗНАЧИТ, ТЕБЕ □";D;"□ ДНЕЙ!»
70 END
```





Поразвлекайся

1. Напиши программу, которая вычислит твой возраст в месяцах.

2. Составь программу, которая вычислит, сколько часов в своей жизни ты провел во сне.

3. Расставь инструкции следующей программы в нужном порядке.

```
SCREEN 0:KEY OFF:NEW
INPUT A
END
INPUT B
?«СКОЛЬКО ВСЕГО МАЛЬЧИКОВ?»
P=B/A*100
REM ВЫЧИСЛЕНИЕ ПРОЦЕНТОВ
REM ВЫЧИСЛЕНИЕ ПРОЦЕНТА МАЛЬЧИКОВ
REM В КЛАССЕ
?«СКОЛЬКО ВСЕГО УЧЕНИКОВ В
  ТВОЕМ КЛАССЕ?»
?«ПОЛУЧАЕТСЯ □»;P; «□ ПРОЦЕНТОВ
  МАЛЬЧИКОВ»
REM ПЕРЕМЕННЫЕ: A=ОБЩЕЕ ЧИСЛО
                    УЧЕНИКОВ
REM                 B=ЧИСЛО МАЛЬЧИКОВ
REM                 P=ПРОЦЕНТ МАЛЬЧИКОВ
```

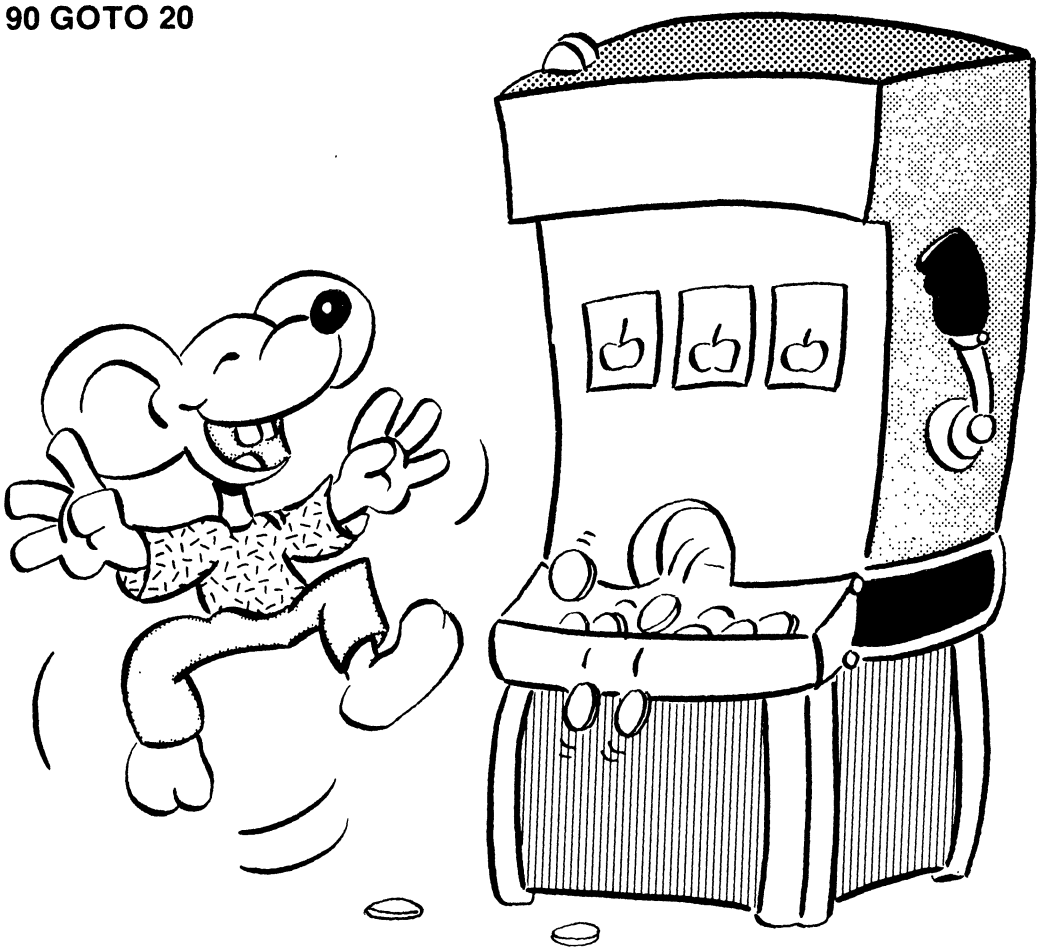
4. Вычисли число девочек в твоем классе (в процентах от общего числа учеников).

5. К какой-нибудь из программ, составленных ранее, добавь комментарии.

СЛУЧАЙНЫЕ ЧИСЛА

Ты можешь сделать свои программы более увлекательными, если введешь в них элемент случайности. Вот один из возможных примеров:

```
SCREEN 0:KEY OFF:NEW
10 REM **** ЗАГАДОЧНОЕ ЧИСЛО ****
20 X=INT(100*RND(1))+1
30 ? «ПОДОЖДИ НЕМНОЖКО, ПОКА Я
      НАЙДУ ЧИСЛО»
40 ? «УГАДАЙ, КАКОЕ ЧИСЛО Я ЗАДУМАЛ?»
50 INPUT G
60 IF G < X THEN PRINT «СЛИШКОМ МАЛО!» : GOTO 40
70 IF G > X THEN PRINT «СЛИШКОМ МНОГО!»: GOTO 40
80 IF G = X THEN PRINT «ВОТ ЭТО ДА! ТОЧНО!»
90 GOTO 20
```

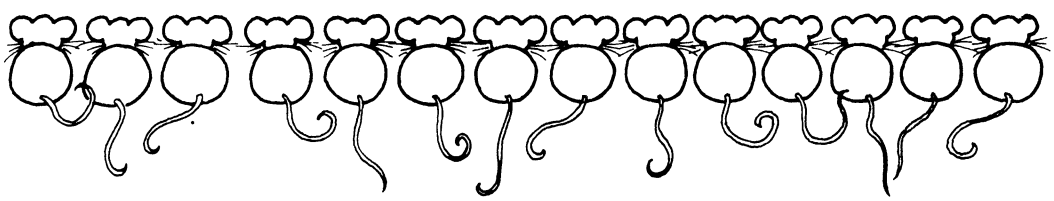


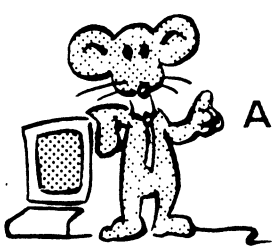
Для того чтобы компьютер выдавал случайные числа, тебе следует воспользоваться инструкцией такого типа:

```
X=INT(100*RND(1))+1
```

Имя переменной можешь выбрать по своему усмотрению.

Это число определяет верхнюю границу выдаваемых чисел. Если ты хочешь, чтобы компьютер создавал случайные числа от 0 до 10, ты должен вместо 100 написать 10.





А теперь поразвлекайся

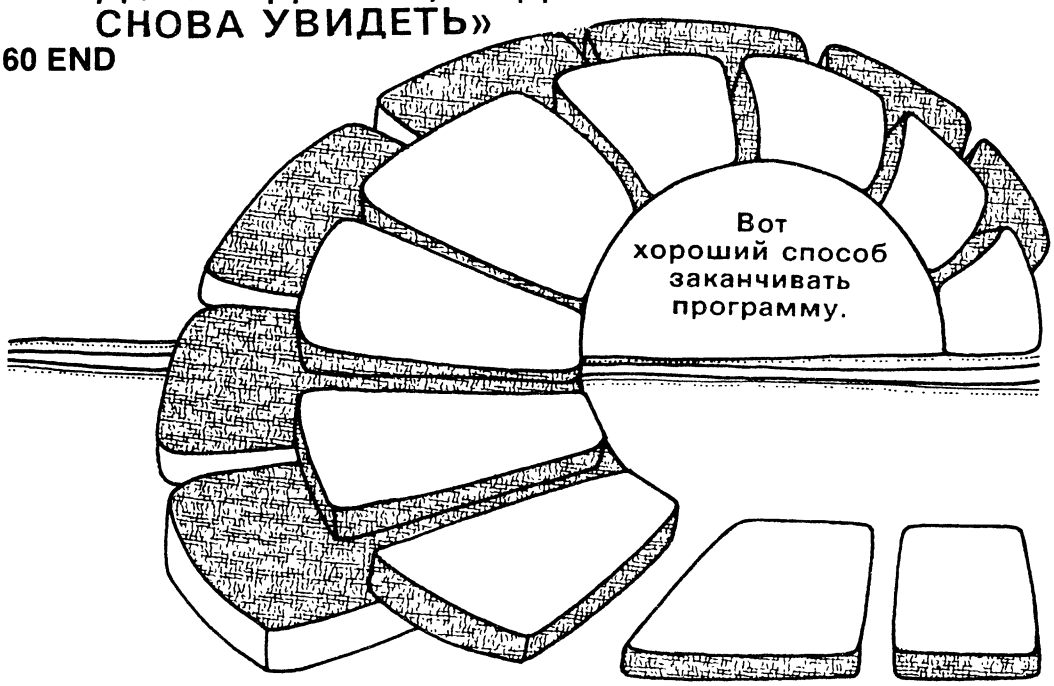
1. Попробуй изменить программу «Загадочных чисел». Сумеешь сделать ее более трудной? А наоборот, более легкой?

2. Предложи компьютеру выдать 10 случайных чисел от 1 до 20.

РАЗВЕТВЛЕНИЯ НА НЕСКОЛЬКО НАПРАВЛЕНИЙ

Комбинируя случайные числа, ты можешь заставить компьютер разнообразить ответы. Посмотри на эту программу:

```
SCREEN 0:KEY OFF:NEW
10 REM *** ИГРА В УМНОЖЕНИЕ ***
20 A=INT(12*RND(1))+1
30 B=INT(12*RND(1))+1
40 ? A;"□X□";B;"□=□";
50 INPUT C
60 IF C < A*B THEN PRINT «МАЛО!» : GOTO 40
70 IF C > A*B THEN PRINT «МНОГО!» : GOTO 40
80 R=INT(3*RND(1))+1: ON R GOTO 90, 100, 110
90 ? «ПОЗДРАВЛЯЮ» : GOTO 120
100 ?«БРАВО! ПОПАЛ В ТОЧКУ!» : GOTO 120
110 ?«ПРОДОЛЖАЙ, ТЫ НА ПРАВИЛЬНОМ
                                    ПУТИ»
120 ?«ХОЧЕШЬ ПОИГРАТЬ ЕЩЕ (ДА ИЛИ НЕТ)?»
130 INPUT Z$
140 IF Z$=«ДА» THEN 20
150 ? «ДО СВИДАНИЯ, НАДЕЮСЬ СКОРО ТЕБЯ
      СНОВА УВИДЕТЬ»
160 END
```



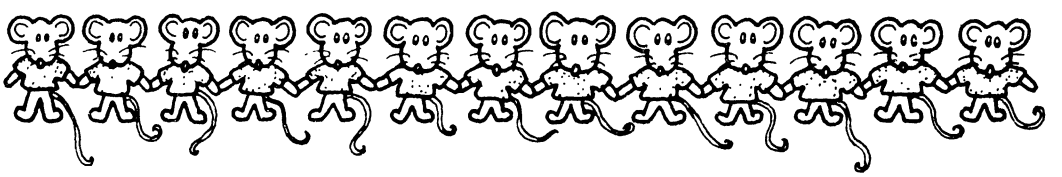
Инструкция, управляющая разветвлениями на несколько направлений, имеет вид

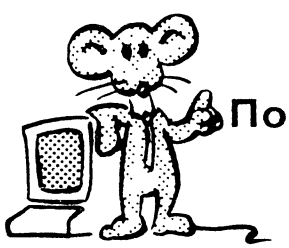
ON X GOTO 90, 100, 110

Числовая переменная.	Номера строк программы, к которым хочешь перейти.

Эта инструкция поможет тебе сэкономить немало времени. Действительно, если бы ее не было, тебе пришлось бы писать вот столько:

```
81 IF X=1 THEN 90
82 IF X=2 THEN 100
83 IF X=3 THEN 110
```





Поразвлекайся

1. Измени программу умножения так, чтобы ты сумел проверить, как усвоил деление.

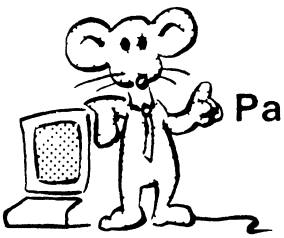
2. Добавь сообщения и измени строку 80 по смыслу.

ТАБЛИЦЫ

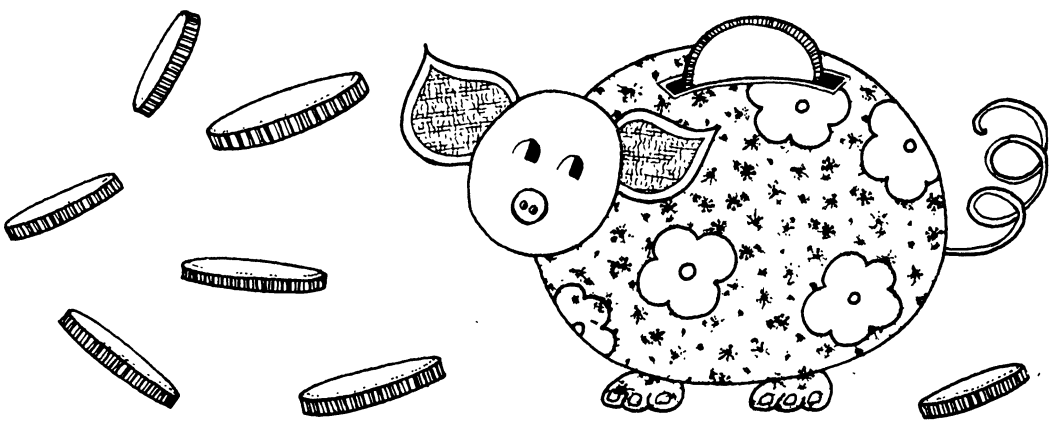
Часто бывает так, что какие-то данные удобно представлять в виде таблицы. Например, если ты хочешь составить список бутербродов, которые надо приготовить для гостей в зависимости от их числа, то данные ты будешь обязательно заносить в столбцы, т. е. составлять таблицу.

```
SCREEN 0:KEY OFF:NEW
10 REM *** ПИКНИК В ЧЕСТЬ ОКОНЧАНИЯ
           УЧЕБНОГО ГОДА ***
20 REM ПЕРЕМЕННЫЕ: E= ЧИСЛО УЧЕНИКОВ
30 REM               S= ЧИСЛО БУТЕРБРОДОВ
40 ? «УЧЕНИКОВ □□□□";"□□□□ БУТЕРБРОДОВ»
50 ?
60 E=1
70 S=E*4
80 ? E, S
90 E=E+1
100 IF E < 21 THEN 70
110 END
```

Экран твоего компьютера разделен на три зоны, называемые *колонками*. Внимание! Не путай эти колонки со столбцами, используемыми при рисовании по координатной сетке.



Развлекись

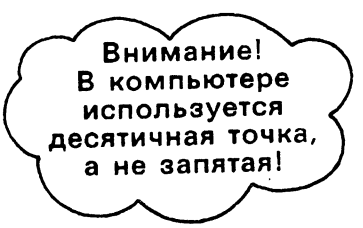


1. Составь программу, с помощью которой ты мог бы заносить в таблицу определенное значение суммы денег, откладываемой в течение года, по одному разу в неделю, каждый раз одинаковой. Сумма должна накапливаться.

2. Составь таблицу перевода миль в километры (или, наоборот, – километров в мили):

1 миля = 1,609 км;

1 км = 0,621 мили.



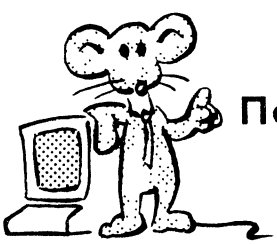
ДАННЫЕ

До сих пор ты пользовался только двумя способами ввода данных в программу. Сейчас ты увидишь, что существует еще и третий:

```
SCREEN 0:KEY OFF:NEW
10 REM *** ПОД-МИ-ГИ-ВАТЬ-ПОД-МИ-ГИ-ВАТЬ ***
20 CLS
30 ? «ЧИТАЙ СЛОВА, КОТОРЫЕ МИГАЮТ
      НА ЭКРАНЕ»
40 ? «ТЫ ХОЧЕШЬ, ЧТОБЫ ОНИ МИГАЛИ
      С КАКОЙ ЧАСТОТОЙ?»
50 ? «ВЫБЕРИ ЧИСЛО 100, 200, 300 и т. д.»
60 ? «ЧЕМ БОЛЬШЕ БУДЕТ ВЫБРАННОЕ
      ЧИСЛО,»
70 ? «ТЕМ МЕНЬШЕ БУДЕТ СКОРОСТЬ
      МЕЛЬКАНИЯ СЛОВ.»
80 INPUT X
90 CLS
100 FOR C=1 TO 500:NEXT C
110 FOR A=1 TO 8
120 READ A$
130 ?A$
140 FOR C=1 TO X:NEXT C
150 CLS
160 FOR C=1 TO 500:NEXT C
170 NEXT A
180 DATA МАШИНА, ДОМ, ДЕРЕВО, ЛИСТОК
190 DATA ЦВЕТОК, ШКОЛА, ДЕНЬГИ, МОНЕТА
200 END
```

Инструкция **DATA** служит для создания списка данных внутри программы, а инструкция **READ** требует от компьютера чтения данных из этого списка. Видишь, имеются два ключевых слова, которые неразделимы и действуют всегда обязательно вместе.

Слово **DATA** не является исполняемой инструкцией. Это слово можно помещать в любом месте программы по твоему желанию. Но все же я советую тебе помещать его в самом конце программы, перед словом **END**, чтобы его легко можно было выделить среди остальных.



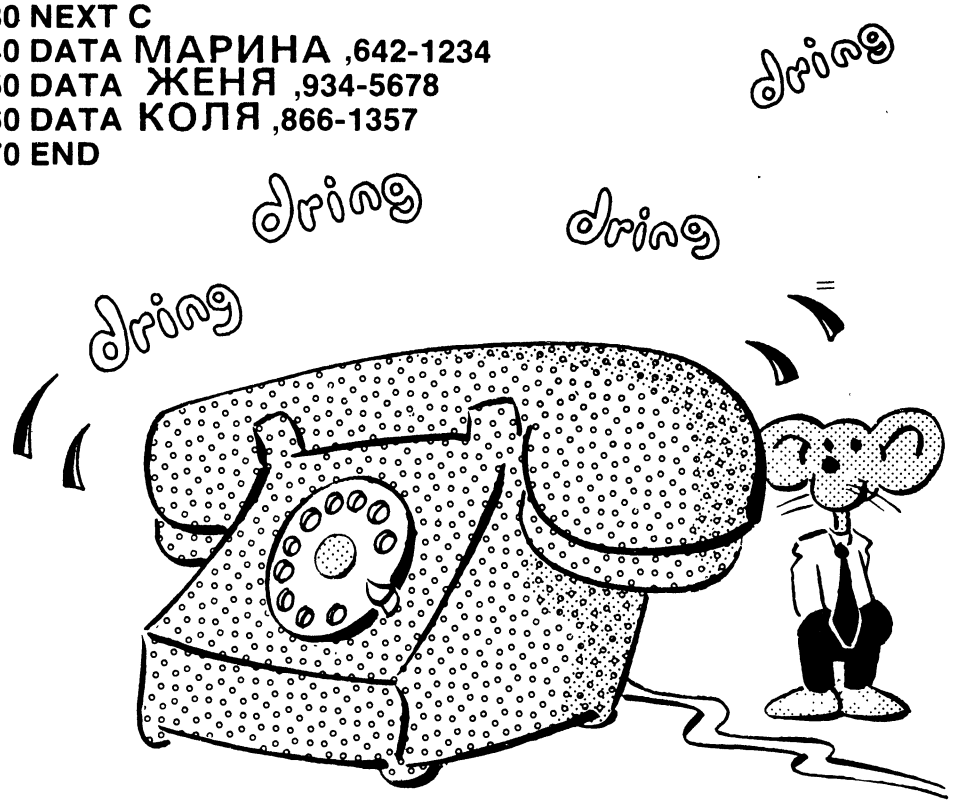
Поразвлекайся

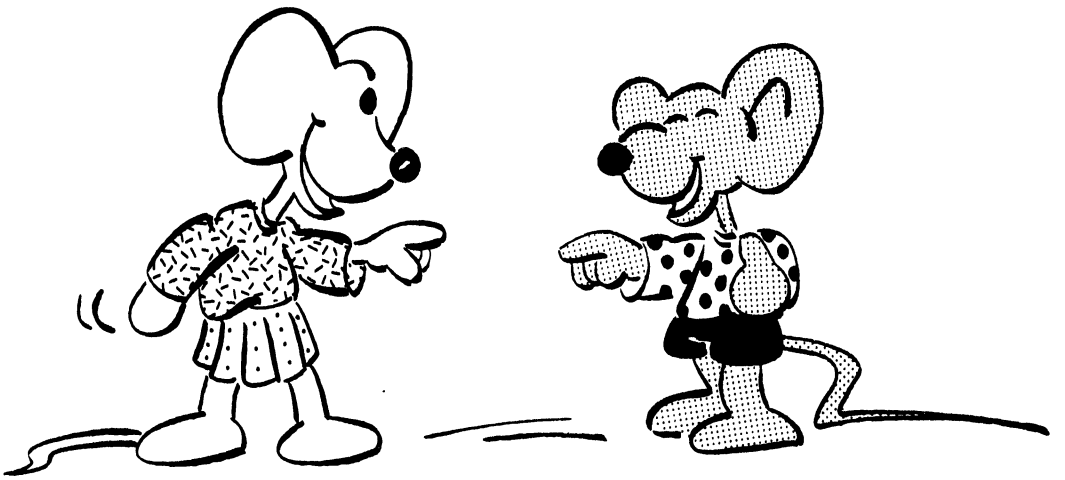
1. Замени слова, обозначенные в инструкции **DATA**, на другие.
2. Напиши программу, по которой компьютер читал бы числовые данные, указанные в списке **DATA**.

ОТОБРАЖЕНИЕ ТАБЛИЦЫ

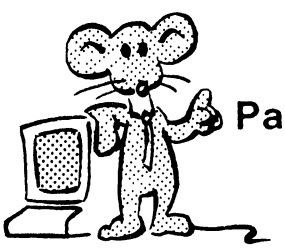
Для отображения данных в столбцах в виде таблицы ты можешь использовать также функцию TAB.

```
SCREEN 0:WIDTH 40:NEW
10 REM ***НОМЕРА ТЕЛЕФОНОВ ***
20 REM ПЕРЕМЕННЫЕ: N$=ИМЯ
30 REM                T$=НОМЕР
40 ?TAB(7)«НОМЕР ТЕЛЕФОНА»
50 ?TAB(7)"==========="
60 ?:?:?
70 ?TAB(5) «ИМЯ» ;TAB(20) «НОМЕР»
80 ?TAB(5)"———";TAB(20)"———————"
90 ?
100 FOR C=1 TO 3
110 READ N$,T$
120 ?TAB(5)N$;TAB(20)T$
130 NEXT C
140 DATA МАРИНА ,642-1234
150 DATA ЖЕНЯ ,934-5678
160 DATA КОЛЯ ,866-1357
170 END
```





Функция **TAB**, действующая только совместно с инструкцией **PRINT**, помещает курсор в точку, имеющую координаты, указанные в скобках.



Развлекись

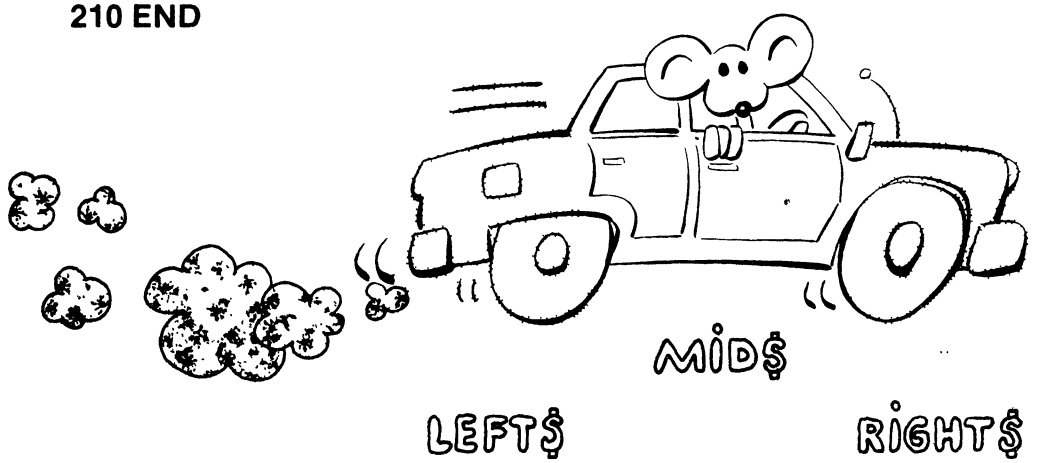
1. Замени данные в списке **DATA** предыдущей программы так, чтобы на их месте оказались записаны настоящие имена и номера телефонов твоих друзей.
2. Изобрази вот такой рисунок в центре экрана:

```
= = = = = = = = = = = = = = = = = = =

          + + + + + + + +
          + + + + + + + +
          + +         + +
          + +         + +
          + + + + + + + +
          + + + + + + + +

= = = = = = = = = = = = = = = = = = =
```

СИМВОЛЬНЫЕ СТРОКИ

Любая последовательность букв или символов, заключенная в кавычки, представляет собой символьную строку, а любая ее часть называется подстрокой. Соединяя различным образом отдельные части строки, можно изменять порядок следования символов.

```
SCREEN 0:KEY OFF:NEW
10 REM *** ИГРА В АНАГРАММЫ ***
20 ? «НАЙДИ ПЯТЬ СПРЯТАННЫХ СЛОВ»
30 FOR A=1 TO 6
40 READ B$
50 IF B$= «ЗВОНОК» THEN 210
60 C$=RIGHT$(B$,2)+MID$(B$,3,2)+LEFT$(B$,2)
70 ?:?:?
80 PRINT C$
90 ?:?:?
100 INPUT «КАКОЕ СЛОВО СПРЯТАНО?»;X$
110 IF X$=B$ THEN 150
120 ? «НЕТ! ПОПЫТАЙСЯ ЕЩЕ РАЗ»
130 FOR C=1 TO 500:NEXT C
140 GOTO 80
150 ? «ПОЗДРАВЛЯЮ! ПРИНИМАЙСЯ
      ЗА СЛЕДУЮЩЕЕ»
160 FOR C=1 TO 500:NEXT C
170 CLS
180 NEXT A
190 DATA ДВОРЕЦ, СОБАКА, КОЛПАК
200 DATA КРУЖОК, ТРУБКА, ЗВОНОК
210 END
```

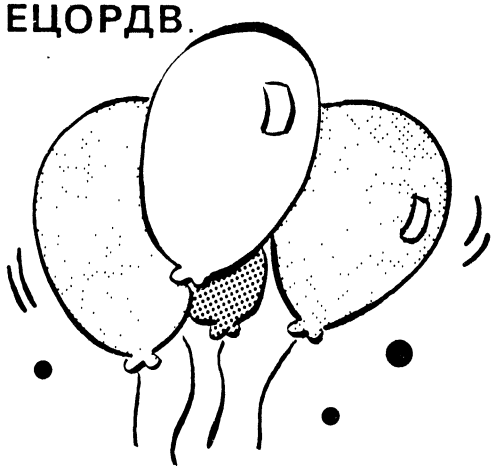


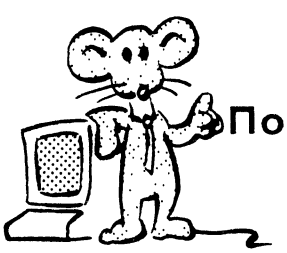
Для образования подстроки ты должен вызвать функции **LEFT$**, **MID$** и **RIGHT$**. В каждом случае необходимо точно указывать, сколько букв должен воспроизвести компьютер.

Чтобы лучше понять, что же при этом происходит, рассмотрим действие предыдущей программы.
Пусть вначале **B$**-«ДВОРЕЦ».

- Функция **RIGHT$(B$, 2)** требует от компьютера отобразить две буквы, расположенные на правом конце строки **B$**, т. е. **ЕЦ**.
- Функция **MID$(B$,3, 2)** требует отобразить две буквы, начиная с третьей буквы строки **B$**, т. е. **ОР**.
- Функция **LEFT$(B$,2)** говорит компьютеру, чтобы он отобразил две буквы, расположенные на левом конце строки **B$**, т. е. **ДВ**.

Поскольку инструкции соединены знаком «+», окончательный результат будет **ЕЦОРДВ**.





Поиграй сам

1. Измени инструкцию в строке 60 так, чтобы слова изменились и образовались другие анаграммы.
2. Придумай новую игру с использованием функций **LEFT$**, **MID$** и **RIGHT$**.

ПРИЛОЖЕНИЯ

А. СЛОВАРЬ ТЕРМИНОВ

Команда – инструкция, которая дается компьютеру в виде простейшего указания.

Отображение – представление на экране текста, изображения или рисунка.

Память оперативная – память, в которой временно хранятся инструкции и данные программы.

Память постоянная – память, в которой хранятся неизменные инструкции.

Переменная – символ, именующий пространство памяти, используемое для хранения элемента информации.

Присваивание – придание переменной конкретного значения.

Программа – последовательность инструкций, предназначенных для компьютера и написанных на языке, который машина понимает или может перевести.

Сообщение об ошибке – примечание, отображаемое на экране компьютером для того, чтобы сигнализировать пользователю о наличии ошибки в программе.

Строка символов – любая последовательность букв, цифр, пробелов, знаков, символов.

Цикл – неоднократное выполнение последовательности инструкций программы.

Цикл паузы – цикл, в котором не выполняется никаких инструкций. Иногда его называют пустым.

END – ключевое слово, используемое для обозначения конца программы.

FOR/NEXT – ключевые слова, используемые для программирования цикла, который должен быть выполнен заданное число раз.

GOTO – ключевое слово, используемое в инструкции безусловного перехода.

IF...THEN – ключевые слова, используемые в инструкции условного перехода.

INPUT – инструкция, требующая от компьютера подождать ответа пользователя.

LEFT$ – функция, приводящая к выделению некоторого числа левых символов из строки символов.

LET – ключевое слово, используемое иногда в операции присваивания.

LINE – ключевое слово, используемое для рисования линии.

LOCATE – ключевое слово, используемое для того, чтобы задать координаты точки.

MID$ – функция, приводящая к выделению некоторого числа символов из средней части строки символов.

NEW – команда, используемая для очистки оперативной памяти компьютера.

ON...GOTO – инструкция разветвления на несколько направлений.

PRINT – инструкция или команда на отображение результата на экране.

READ/DATA – ключевые слова, используемые для чтения (**READ**) данных (**DATA**) в программе.

REM – инструкция, позволяющая вставить комментарии в программу.

RIGHT$ – функция, приводящая к выделению некоторого числа правых символов из строки символов.

RND – функция, предназначенная для генерирования случайных чисел.

RUN – команда, требующая от компьютера выполнения программы.

SCREEN 0 – команда перехода в текстовый режим.

SCREEN 1 – команда перехода в графический режим.

STEP – ключевое слово, указывающее шаг изменения цикла **FOR/NEXT**.

TAB – функция, используемая в инструкции **PRINT** для перемещения курсора вправо.

Б. СООБЩЕНИЯ ОБ ОШИБКАХ

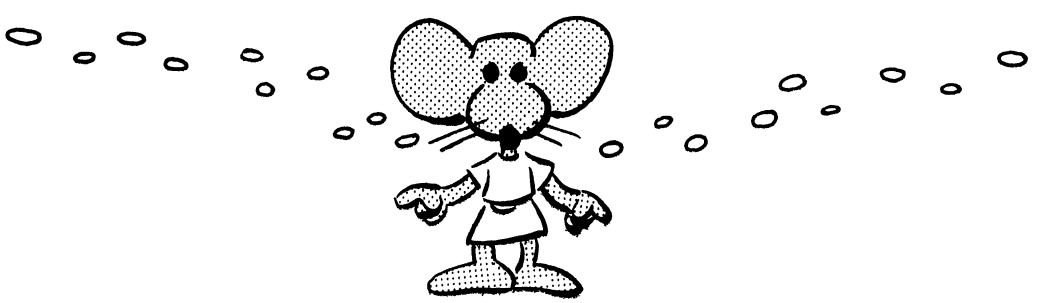
Каждый раз, когда компьютер не понимает, о чем ты его просишь, он отображает на экране сообщение об ошибке. Вот расшифровка основных типов ошибок:

?SYNTAX ERROR – неправильное нажатие клавиши, неудачный выбор переменной, синтаксическая ошибка.

?OUT OF DATA ERROR – недостаточное количество данных, приведенных в команде **DATA**; их должно быть по крайней мере не меньше, чем переменных в инструкции **READ**.

?NEXT WITHOUT FOR ERROR – наличие слова **NEXT** без соответствующего ему слова **FOR** или сопровождение словами **NEXT** и **FOR** различных переменных.

?UNDEF'D STATEMENT ERROR – отсылка к строке с несуществующим номером.



В. ИСПОЛЬЗОВАНИЕ ГИБКИХ ДИСКОВ

Как вставить диск в считывающее устройство (дисковод)

Открой дверцу.
Вставь диск внутрь считывающего устройства.
Закрой дверцу.
Включи компьютер в сеть.

Проделав все это, ты увидишь, как на экране появится дата. Если надо, то напиши новую или нажми на клавишу ENTER (↵).

Затем компьютер отобразит время. Уточни его (если необходимо) и нажми на клавишу **ENTER** (↵).

Наконец на экране ты увидишь текст. Напиши **BASICA** и нажми на клавишу **ENTER** (↵).

Разметка гибкого диска

Вставь рабочий диск в левый дисковод.
Включи компьютер.
Напиши дату и время.

Если появится символ **A >**, напиши **FORMAT** и нажми на клавишу **ENTER** (↵).
Замени рабочий диск на чистый и нажми на клавишу **ENTER** (↵). Когда компьютер спросит тебя, ответь **N** (НЕТ), если ты не хочешь размечать другие диски.
Снова вставь рабочий диск в левый дисковод, напиши **BASICA** и нажми на клавишу **ENTER** (↵).

Сохранение программы

Напиши **SAVE**, затем имя программы и нажми на клавишу **ENTER** (↵).
(Чтобы узнать, надежно ли сохраняется программа, напиши **FILES** и нажми на клавишу **ENTER** (↵)).

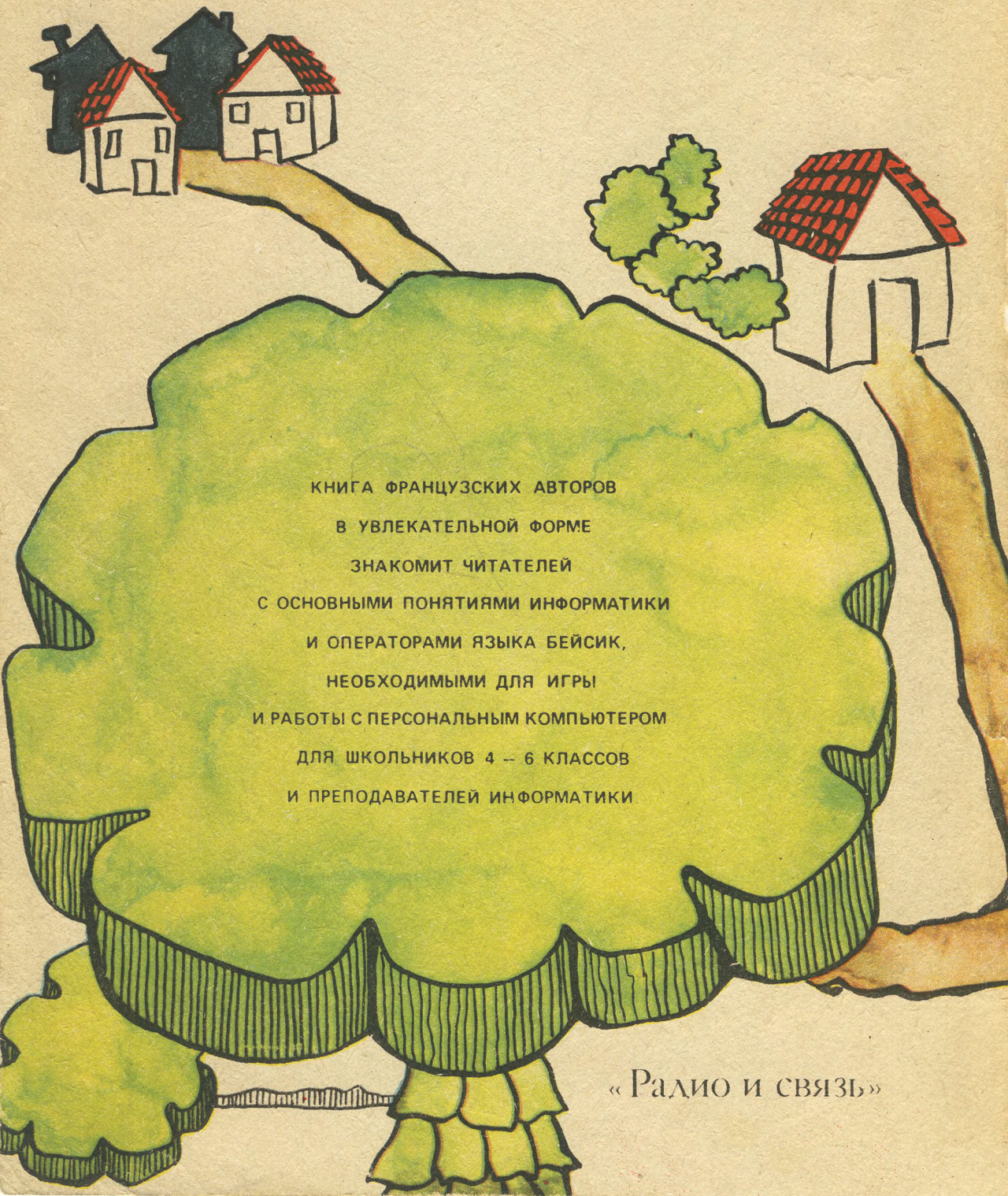
Загрузка программы

Напиши **LOAD**, затем имя программы и нажми на клавишу **ENTER** (↵).
Когда появится курсор, будет ясно, что загрузка закончена.)

Выполнение программы

Напиши **FILES** и нажми на клавишу **ENTER** (↵).
Выбери программу.
Напиши **RUN**, затем имя программы и нажми на клавишу **ENTER** (↵).

1 р.



КНИГА ФРАНЦУЗСКИХ АВТОРОВ
В УВЛЕКАТЕЛЬНОЙ ФОРМЕ
ЗНАКОМИТ ЧИТАТЕЛЕЙ
С ОСНОВНЫМИ ПОНЯТИЯМИ ИНФОРМАТИКИ
И ОПЕРАТОРАМИ ЯЗЫКА БЕЙСИК,
НЕОБХОДИМЫМИ ДЛЯ ИГРЫ
И РАБОТЫ С ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ
ДЛЯ ШКОЛЬНИКОВ 4 – 6 КЛАССОВ
И ПРЕПОДАВАТЕЛЕЙ ИНФОРМАТИКИ

«Радио и связь»